

Forecasting with cohort-based models

Nicolai Vicol

Data Scientist at Wix

November 17, 2021



About WIX

About WIX

Wix offers solutions to **create a professional website** and **manage your business**.

Whether you are an entrepreneur, designer, developer, musician, photographer or restaurant owner, Wix has the right product and features to fit your needs.

Wix products are provided on a **freemium** base.

Wix earns from **premium** upgrades, like domain, storage, bandwidth, online payments, etc.

Wix Premium Plans

Wix gives you 100s of templates, unlimited pages & top grade hosting FREE.

Upgrade to Premium and get even more.

Get Started Today

Website Plans

Great for showcasing a professional site

Business & eCommerce Plans

Essential for accepting online payments

MOST POPULAR

VIP

First Priority Support

€ **26** /month

Unlimited

Entrepreneurs &
Freelancers

€ **14.50** /month

Combo

For Personal Use

€ **8.50** /month

Connect Domain

Most Basic

① This Plan displays Wix ads

€ **4.50** /month

	VIP	Unlimited	Combo	Connect Domain
Custom domain	✓	✓	✓	✓
Free domain for 1 year	✓	✓	✓	—
Remove Wix ads	✓	✓	✓	—
Free SSL certificate	✓	✓	✓	✓
Bandwidth	Unlimited	Unlimited	2GB	1GB
Storage space	35GB	10GB	3GB	500MB
Video hours	5 hours	1 hour	30 minutes	—

200 M

Registered Users

5.5 M

Premium Subscriptions



*as of 2020-12-31, from investor presentation, <https://investors.wix.com/>

The ideas exposed here are **applicable to any company** offering a **freemium** product or a **free trial** of it, which is **free** in a basic or limited setup, but can be **upgraded with premium subscriptions**, for example Spotify, Dropbox, Slack, Mailchimp.



Pick your Premium

Listen without limits on your phone, speaker, and other devices.



1 month free

Individual

\$9.99/month after offer period
1 account

- ✓ Listen to music ad-free
- ✓ Play anywhere - even offline
- ✓ On-demand playback

GET STARTED

Terms and conditions apply. 1 month free not available for users who have already tried Premium.

1 month free

Duo

\$12.99/month after offer period
2 accounts

- ✓ 2 Premium accounts for a couple under one roof
- ✓ Duo Mix: a playlist for two, regularly updated with music you both enjoy
- ✓ Ad-free music listening, play offline, on-demand playback

GET STARTED

Terms and conditions apply. 1 month free not available for users who have already tried Premium.

1 month free

Family

\$15.99/month after offer period
6 accounts

- ✓ 6 Premium accounts for family members living under one roof
- ✓ Family Mix: a playlist for your family, regularly updated with music you all enjoy
- ✓ Block explicit music
- ✓ Ad-free music listening, play offline, on-demand playback
- ✓ Spotify Kids: a separate app made just for kids

GET STARTED

Terms and conditions apply. 1 month free not available for users who have already tried Premium.

1 month free

Student

\$4.99/month after offer period
1 account

- ✓ Hulu (ad-supported) plan
- ✓ SHOWTIME
- ✓ Listen to music ad-free
- ✓ Play anywhere - even offline
- ✓ On-demand playback

GET STARTED

Offer currently includes access to Hulu (ad-supported) plan and SHOWTIME Streaming Service, subject to eligibility. Available only to students at an accredited higher education institution. 1 month free only open to higher education students who haven't already tried Premium. Terms and conditions apply.



Personal		Business		
Plus	Family	Professional	Standard	Advanced
For individuals	For families	For individuals	For smaller teams	For larger teams
Buy now	Buy now	Try for free or purchase now	Try for free or purchase now	Try for free or purchase now

Dropbox core features

Storage	i	2 TB (2,000 GB)	Share 2 TB (2,000 GB)	3 TB (3,000 GB)	5 TB (5,000 GB)	As much space as needed
Users	i	1 user	Up to 6 users	1 user	3+ users	3+ users
Best-in-class sync technology	i	✓	✓	✓	✓	✓
Anytime, anywhere access	i	✓	✓	✓	✓	✓
Computer backup	i	✓	✓	✓	✓	✓
Easy and secure sharing	i	✓	✓	✓	✓	✓
256-bit AES and SSL/TLS encryption	i	✓	✓	✓	✓	✓

<input checked="" type="radio"/> Billed yearly <input type="radio"/> Billed monthly	For individuals \$9.99 / month	For families \$16.99 / month	For individuals \$16.58 / month	For smaller teams \$12.50 / user / month	For larger teams \$20 / user / month
	Buy now	Buy now	Try for free or purchase now	Try for free or purchase now	Try for free or purchase now

Revenue of Wix

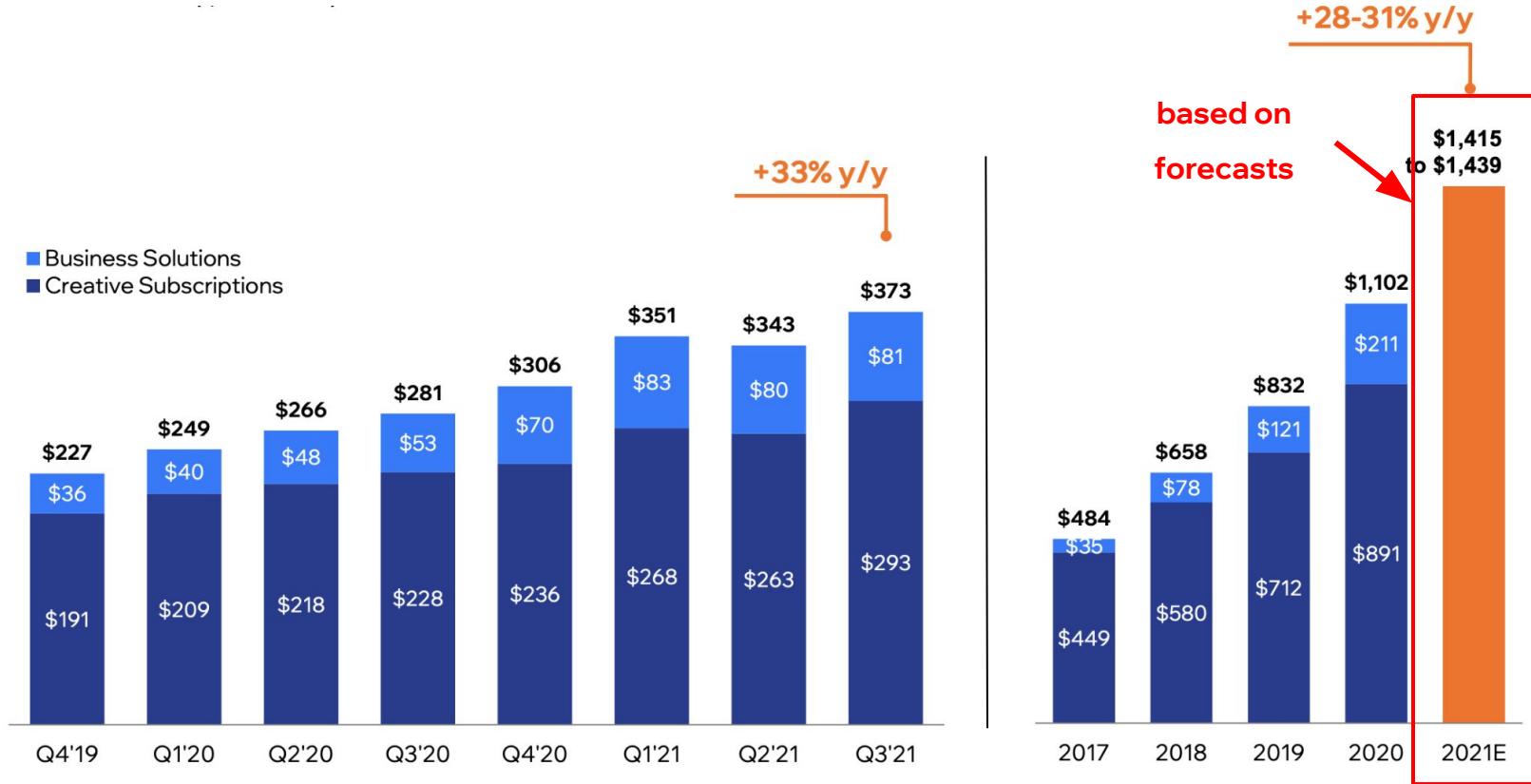
quarterly and yearly



*as of 2021-03-31, from investor presentation, find more: <https://investors.wix.com/>

Revenue of Wix

quarterly and yearly



*as of 2021-03-31, from investor presentation, find more: <https://investors.wix.com/>

Our task as forecasters:

Forecast future incoming cash flows for operational
and strategic planning by management, and also
support the guidance provided to investors.

Revenue of Wix

quarterly and yearly



*as of 2021-11-10, from investor presentation, find more: <https://investors.wix.com/>

Breakdown by main categories

Business Solutions & Creative Subscriptions

Business Solutions

Google Workspace,
Ascend, Ads, TPAs

\$81M

Creative Subscriptions

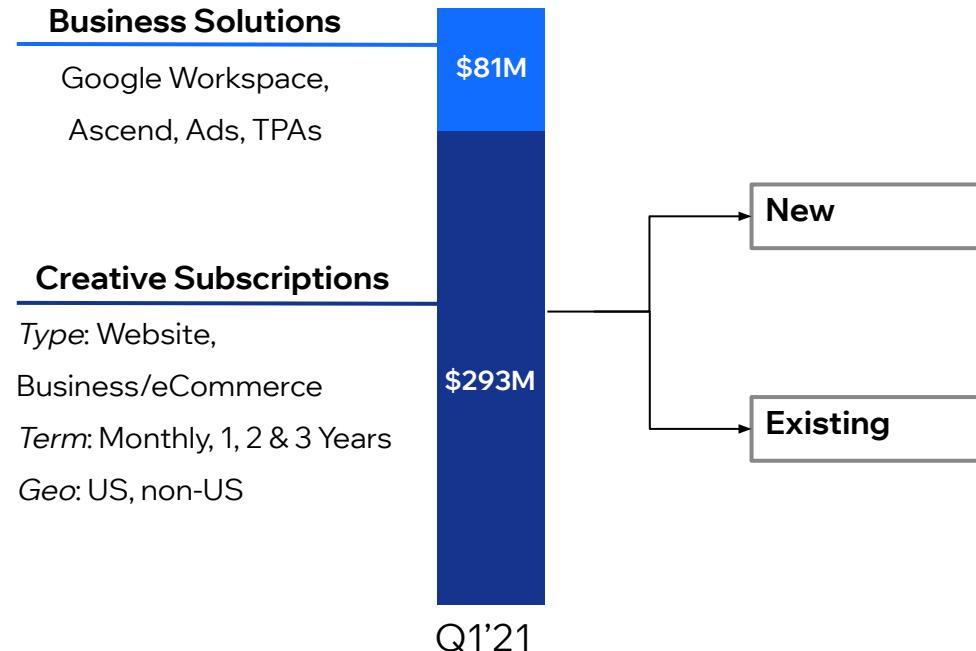
Type: Website,
Business/eCommerce
Term: Monthly, 1, 2 & 3 Years
Geo: US, non-US

\$293M

Q1'21

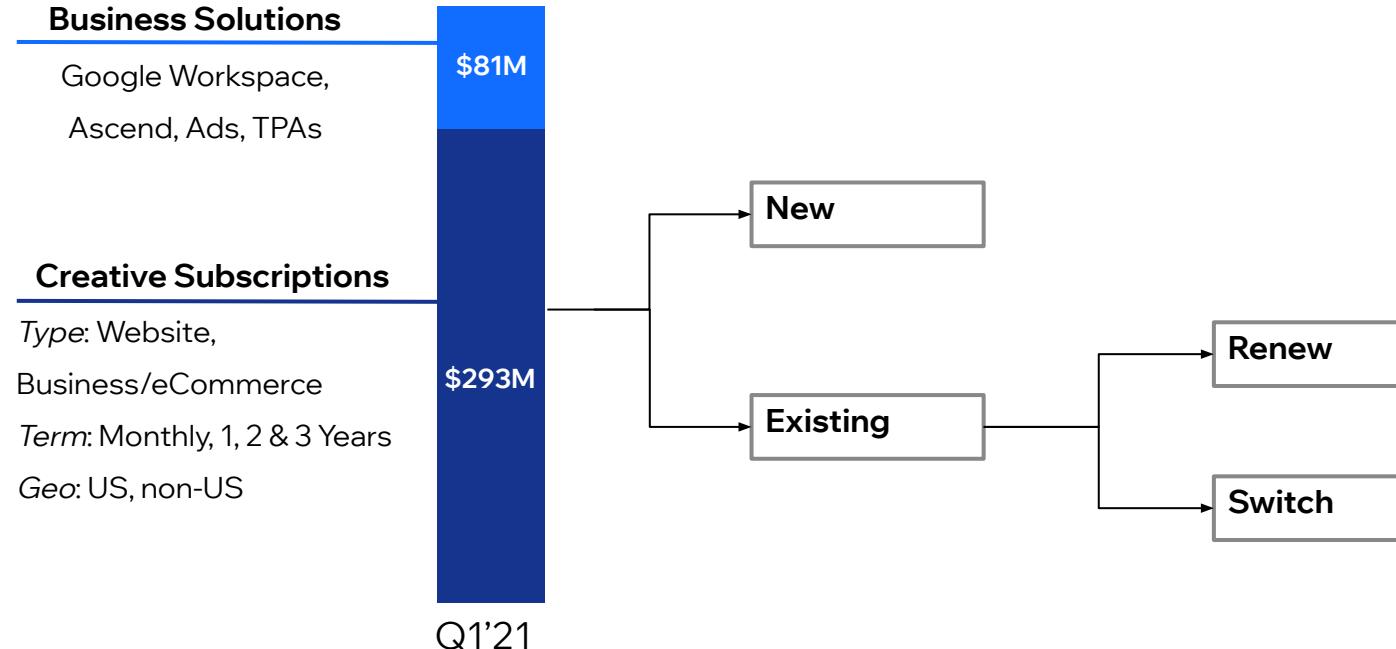
Breakdown by new & existing

New and existing subscriptions

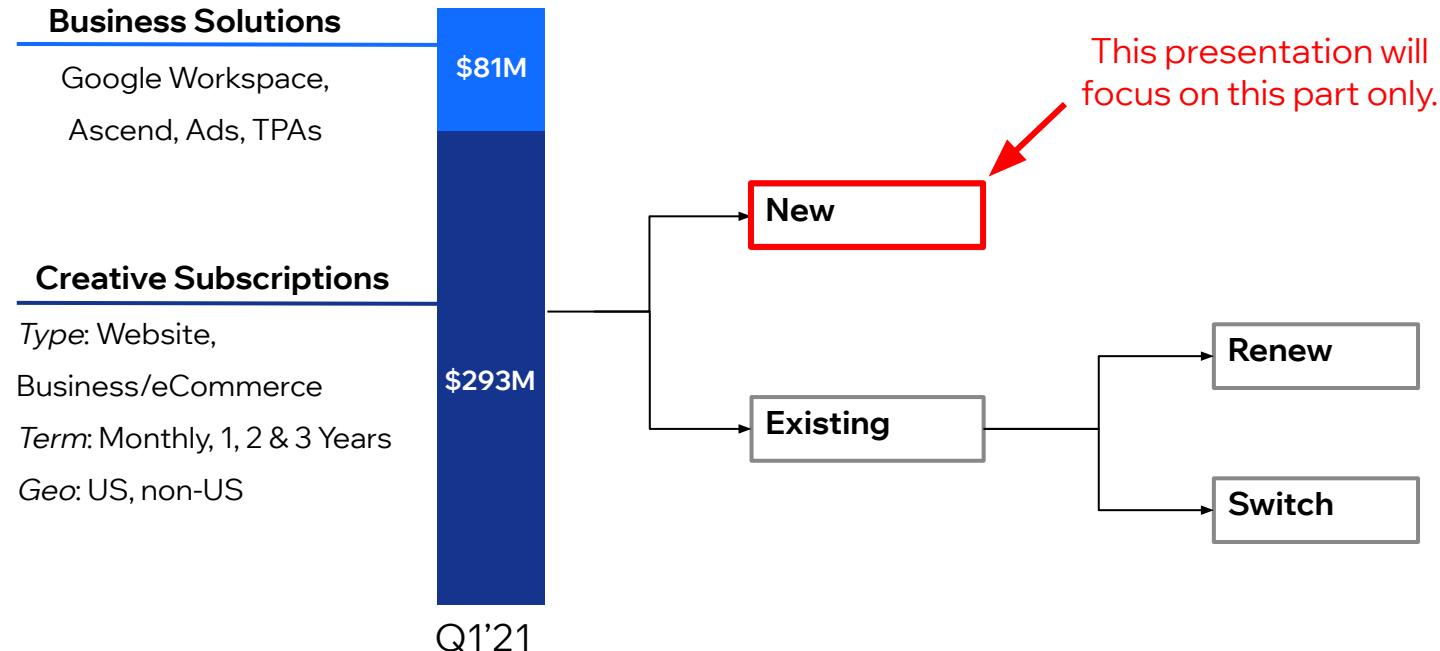


Breakdown of existing subscriptions

Renewed or switched to other plans



Target to forecast: New premium subscriptions



Cohorts of users

What is a cohort?

cohort

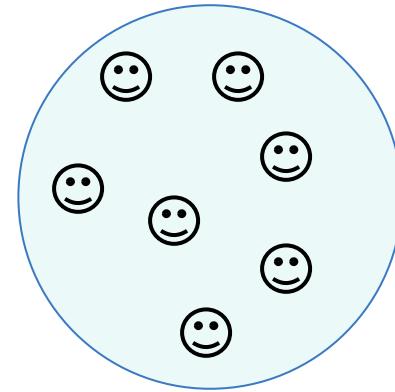
noun [C, + sing/pl verb]

UK /'kəʊ.ho:t/ US /'koʊ.ho:t/

SOCIAL SCIENCE • specialized

a group of people who share a characteristic, usually age:

- *This study followed up a cohort of 386 patients aged 65+ for six months after their discharge home.*



Cohorts at Wix:

users who registered within the same time period

examples

Registered
on
day X

Registered
on
week Y

Registered
on
month Z

Cohorts behaviour and properties

Disclaimer:

This presentation **does not use actual data of Wix**. All charts and numbers in what follows are based on **synthetic data**, generated to **mimic** behaviours and properties that are likely to be observed in users of **other product companies** with a **freemium** business model.

Premiums by upgrade date

for the cohort of registration date: 2019-01-01



Premiums by upgrade date

for the cohort of registration date: 2019-01-01



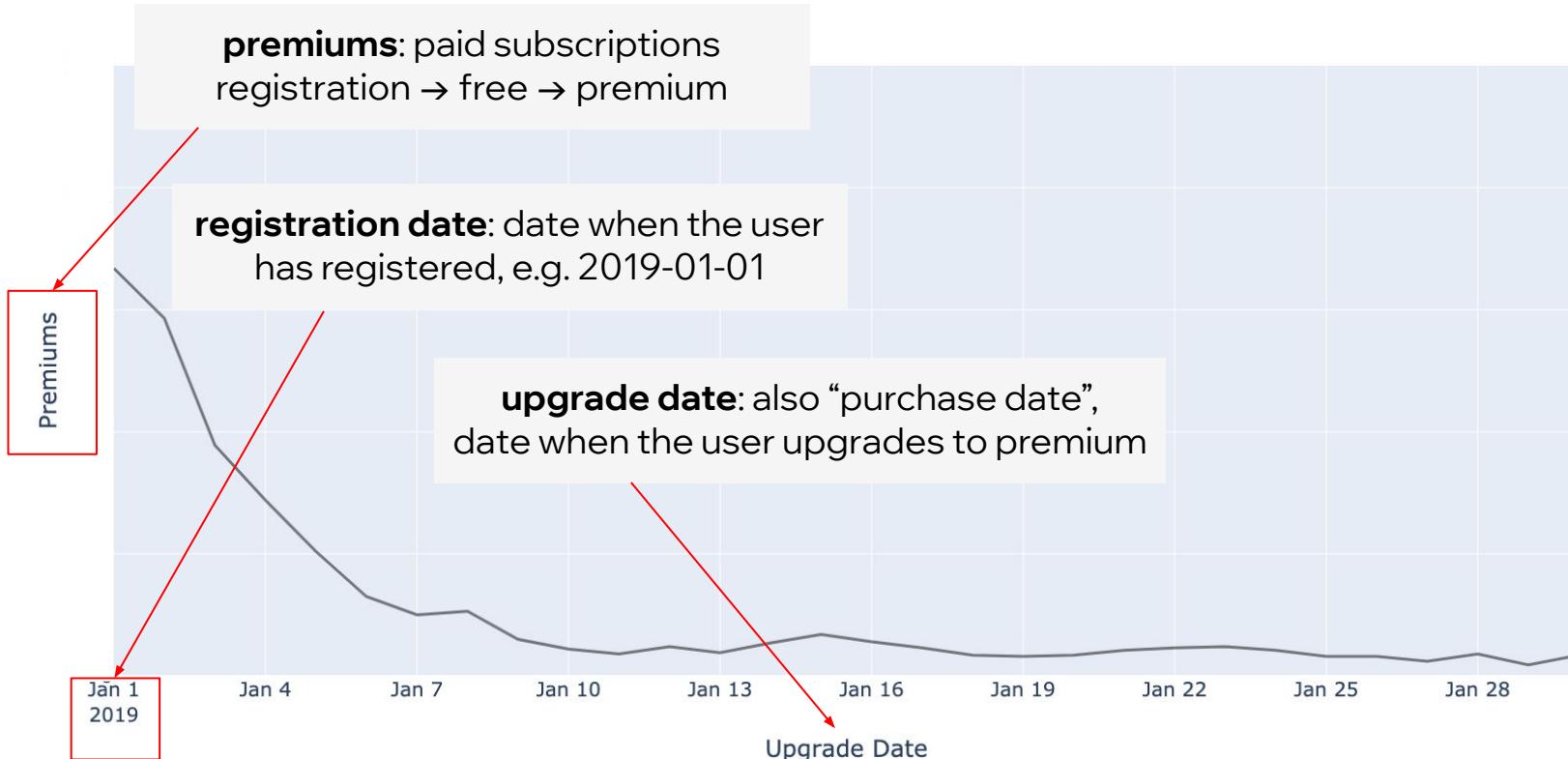
Premiums by upgrade date

for the cohort of registration date: 2019-01-01



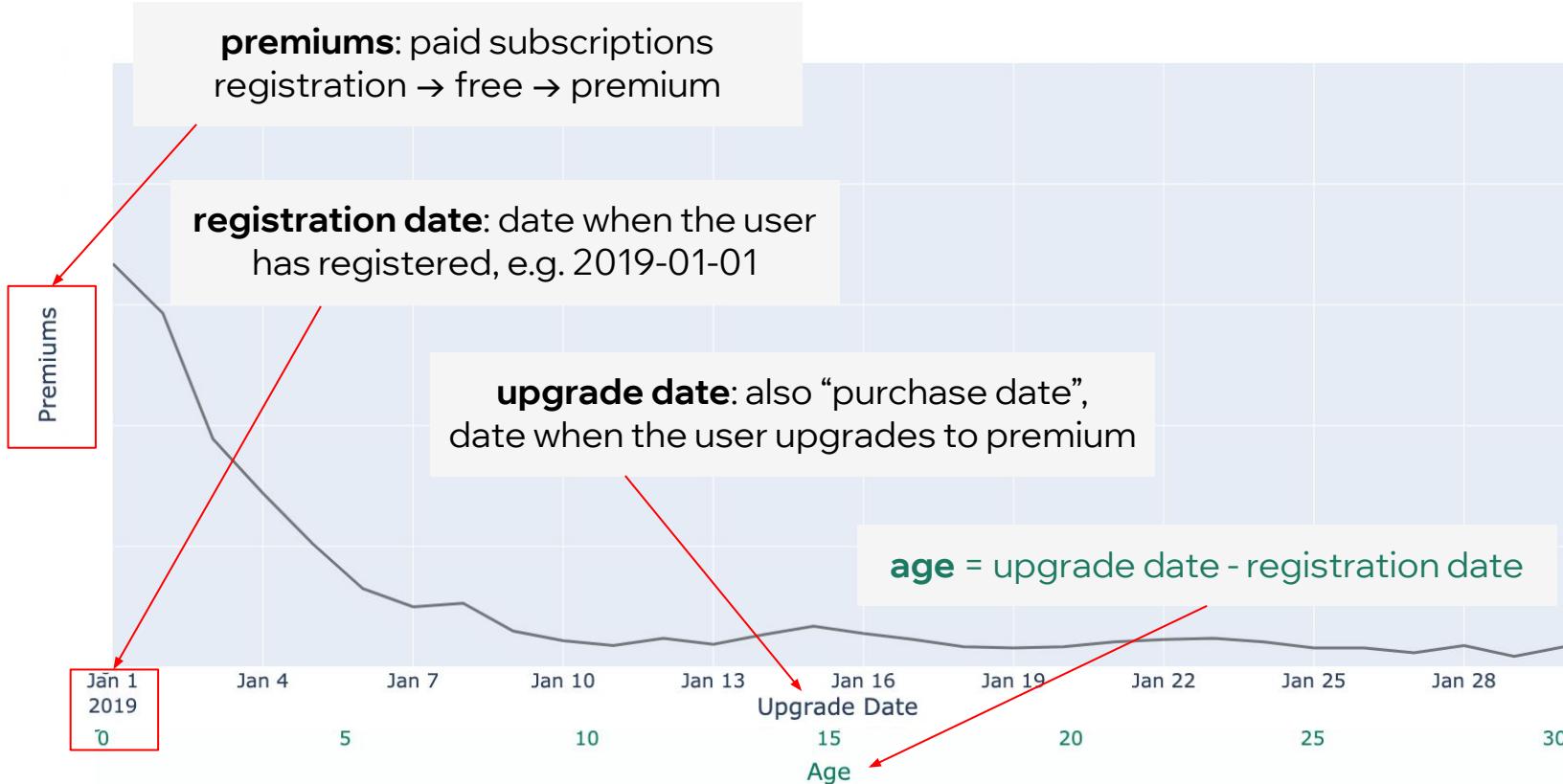
Premiums by upgrade date

for the cohort of registration date: 2019-01-01



Premiums by upgrade date & age

for the cohort of registration date: 2019-01-01



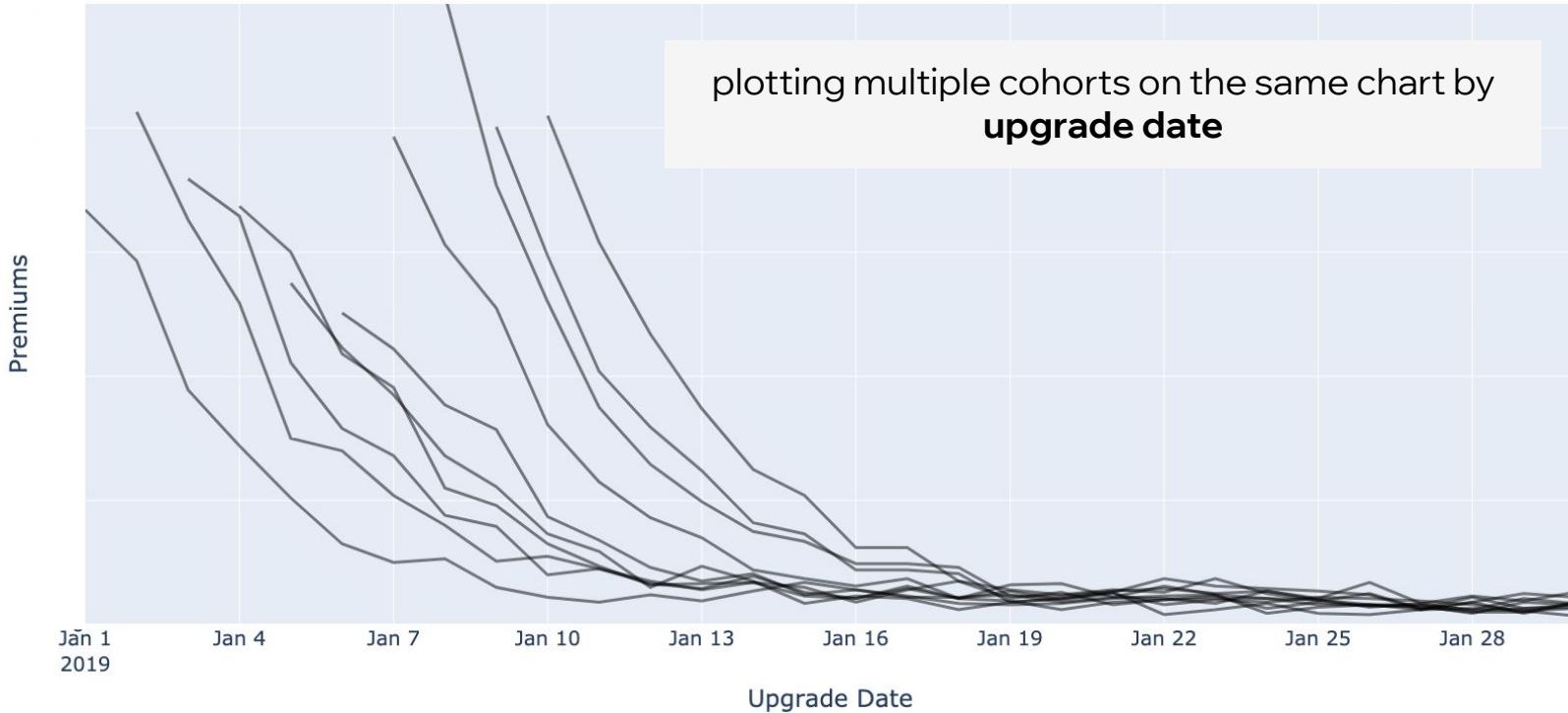
Premiums by upgrade date

for the cohort of registration date: 2019-01-01



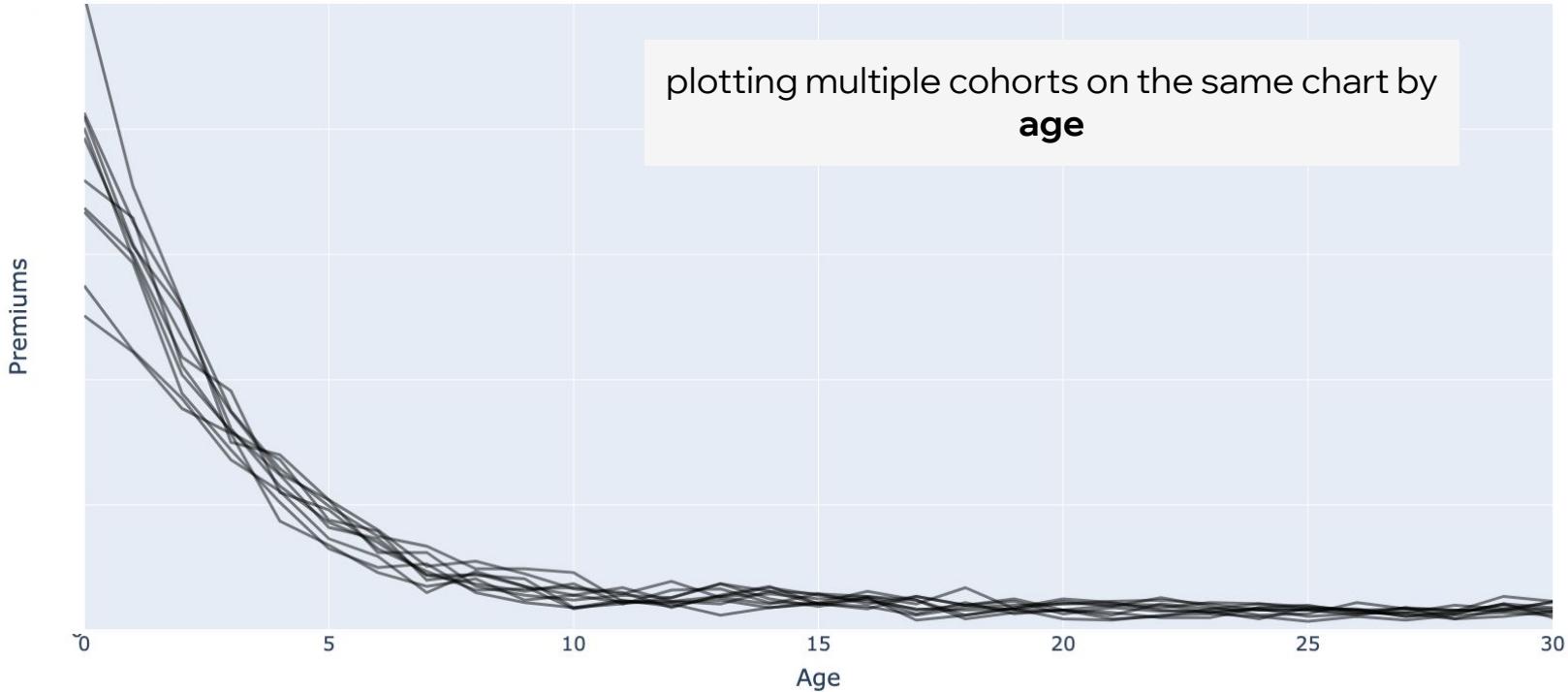
Cohorts behave similarly

they have similar shapes for premiums by upgrade date



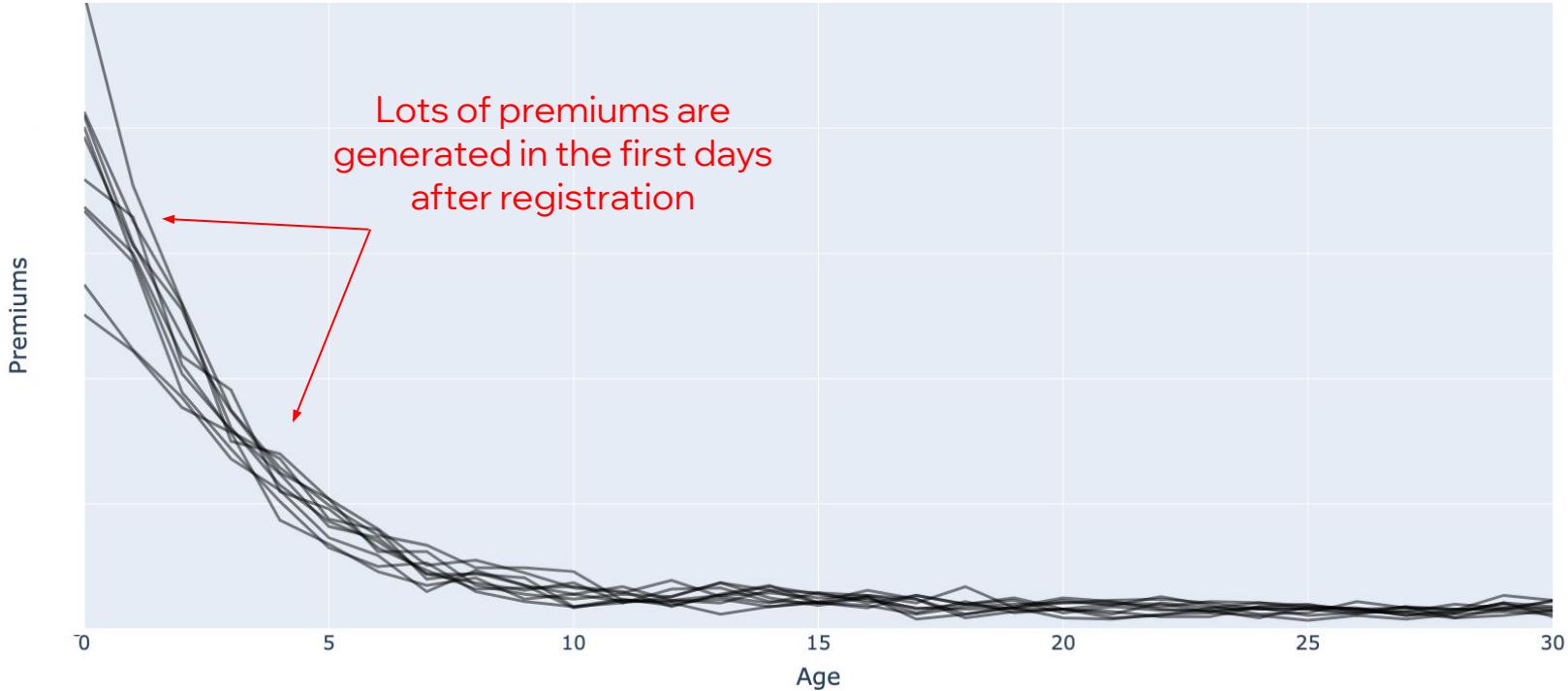
Cohorts behave similarly

they have similar shapes for premiums by age



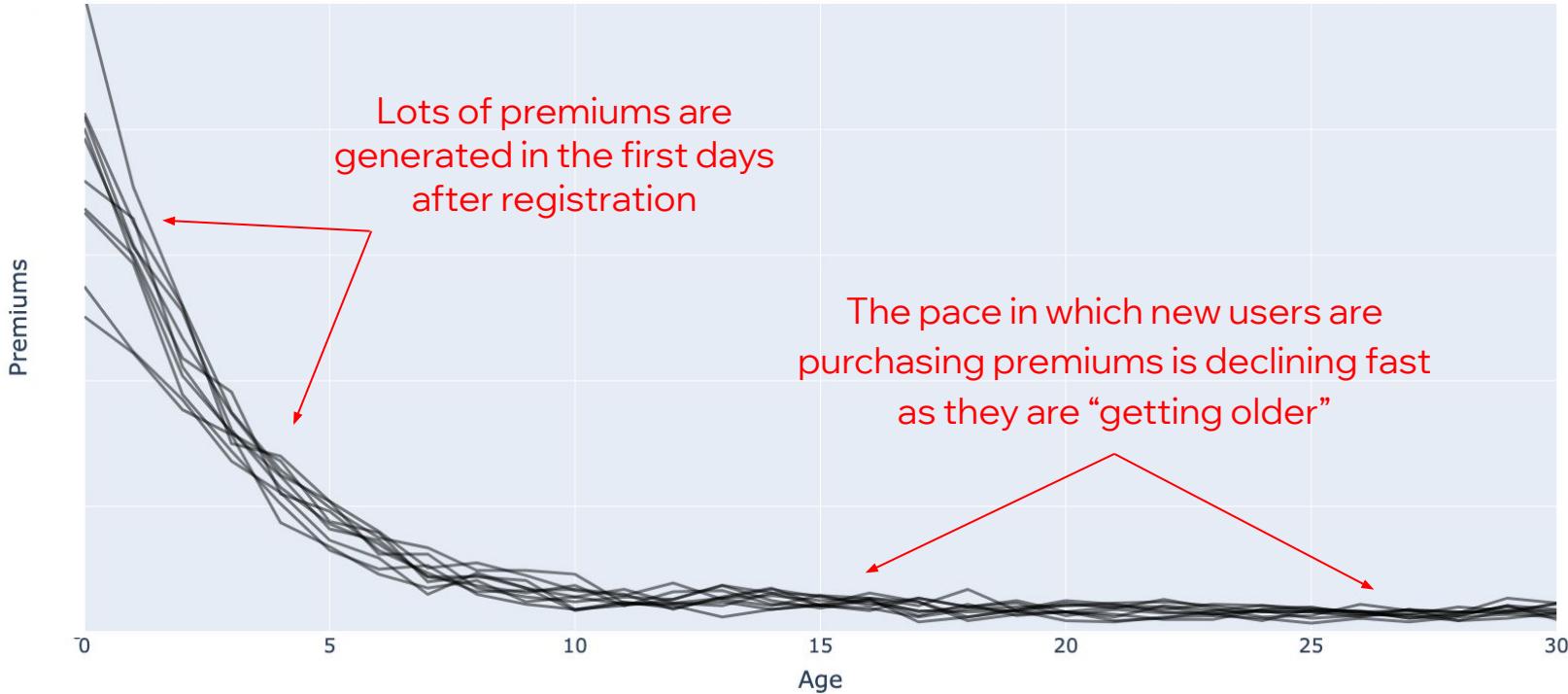
Cohorts behave similarly

they have similar shapes for premiums by age



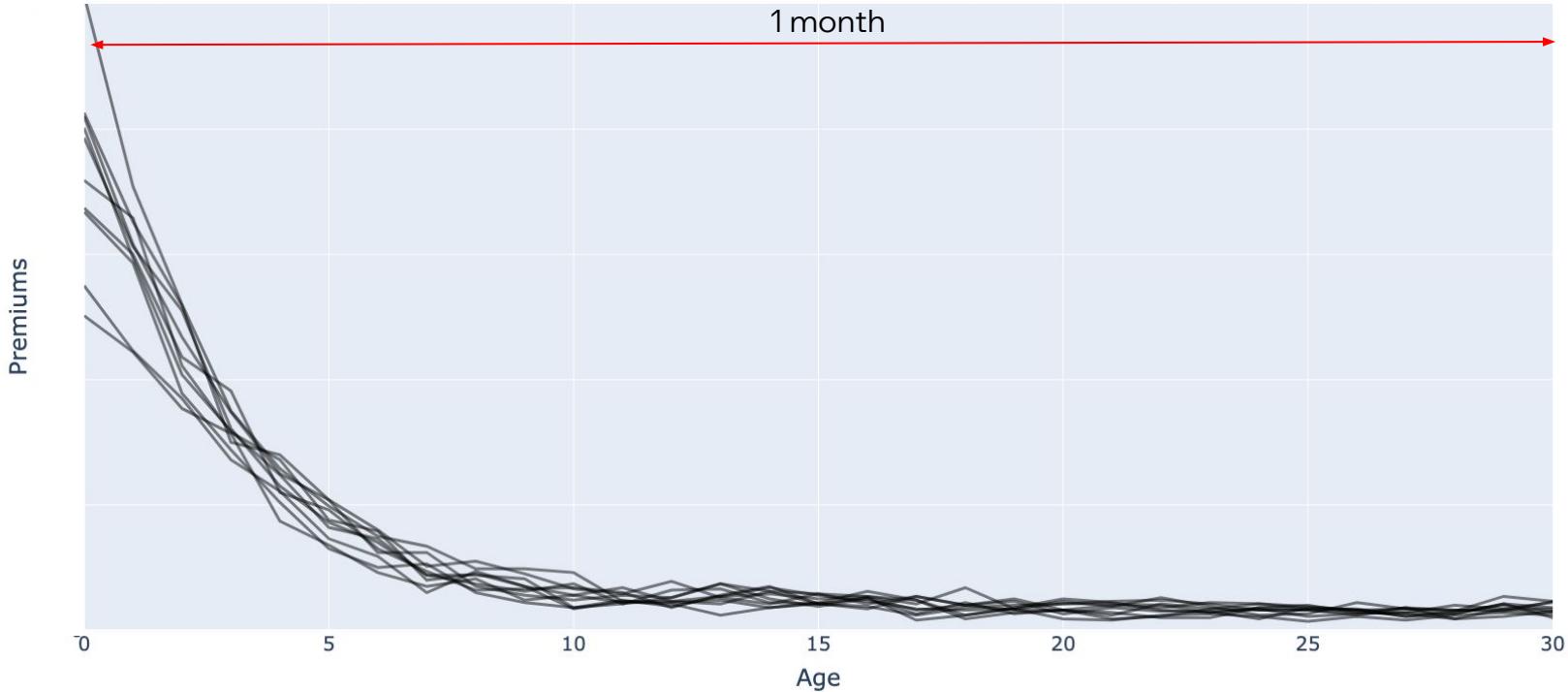
Cohorts behave similarly

they have similar shapes for premiums by age



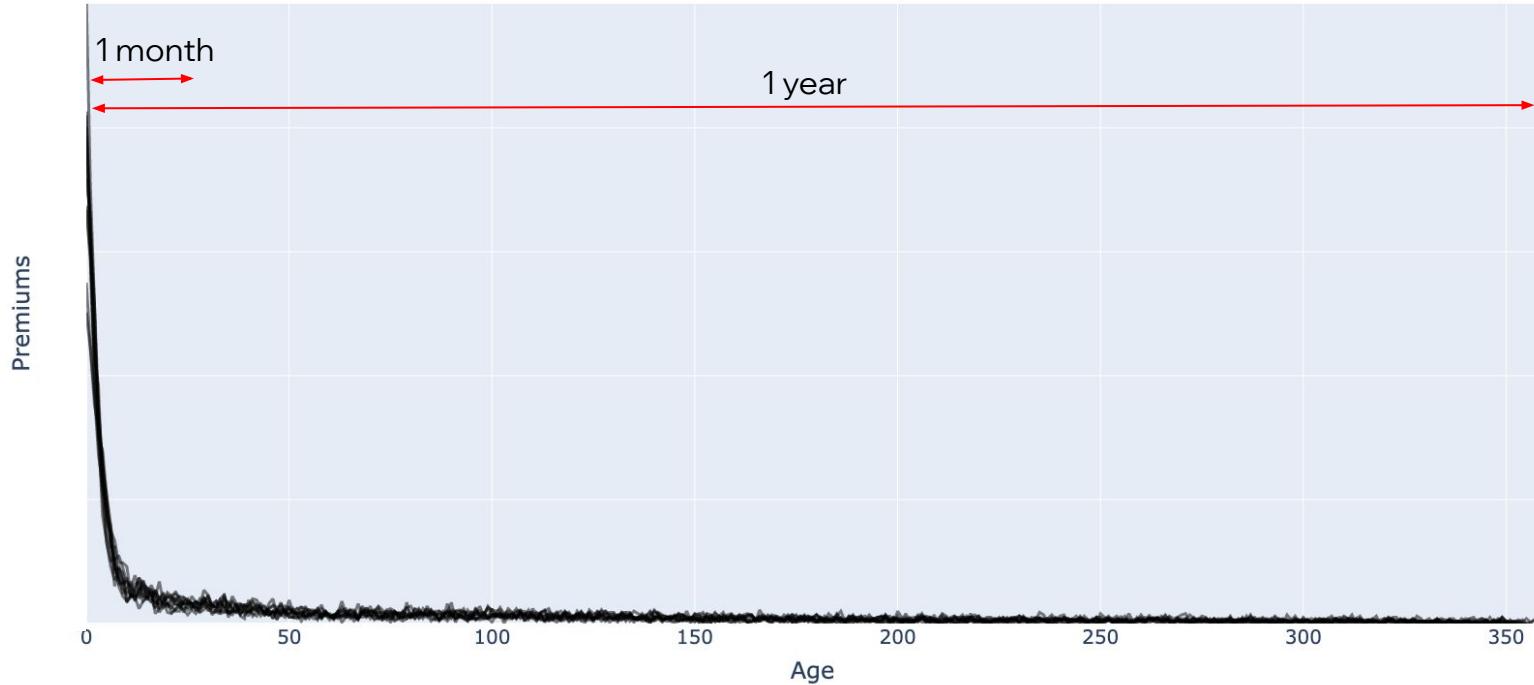
Cohorts behave similarly

they have similar shapes for premiums by age



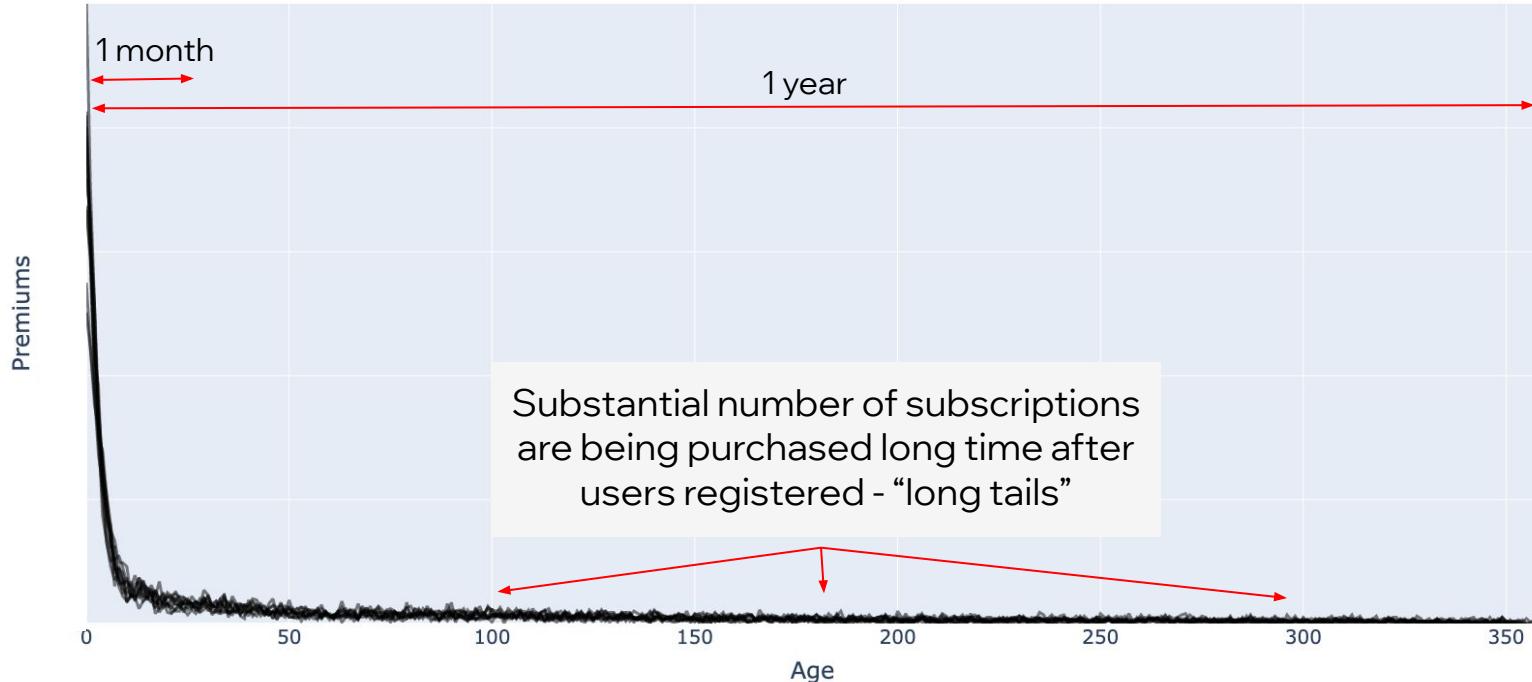
Cohorts have long tails

even for one year ahead they still generate premiums



Cohorts have long tails

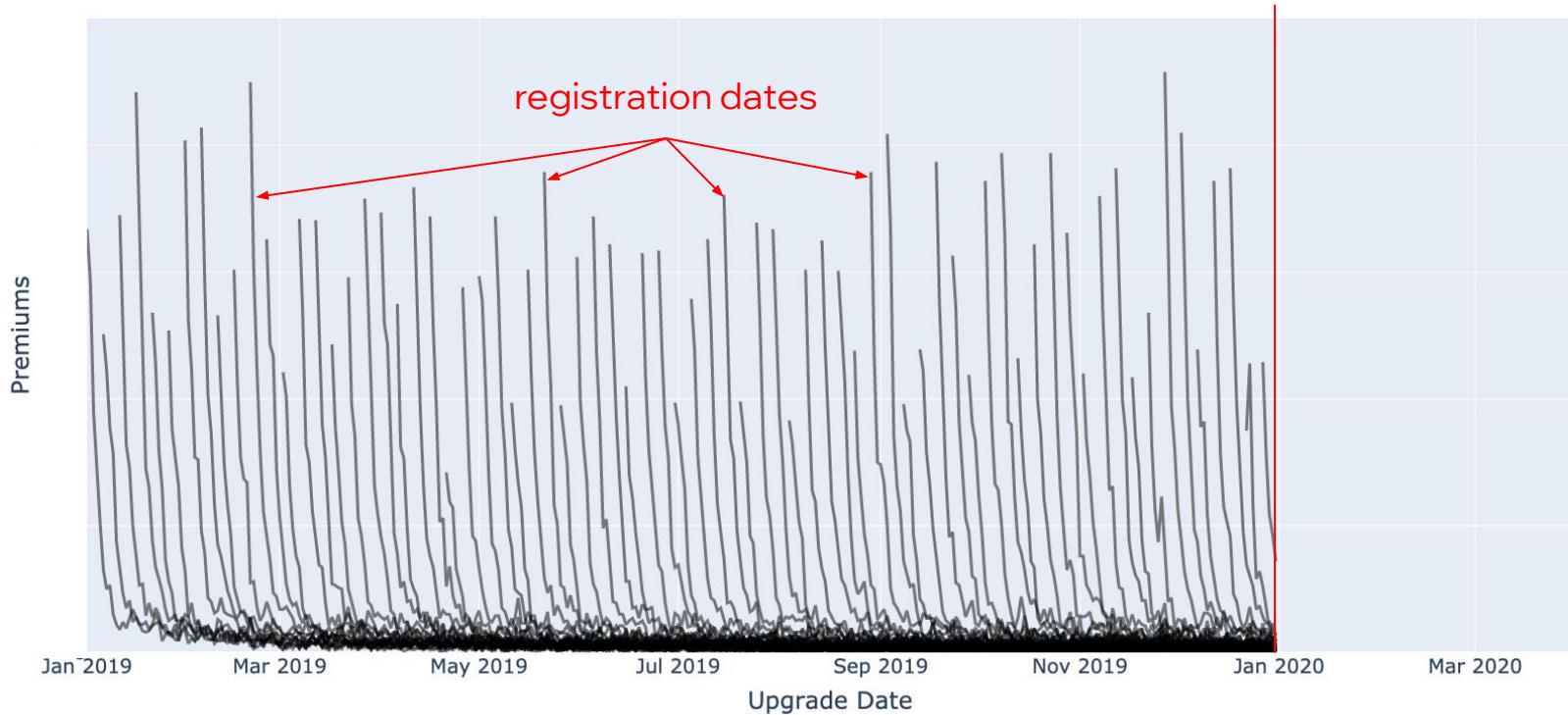
even for one year ahead they still generate premiums



Forecasting cohorts

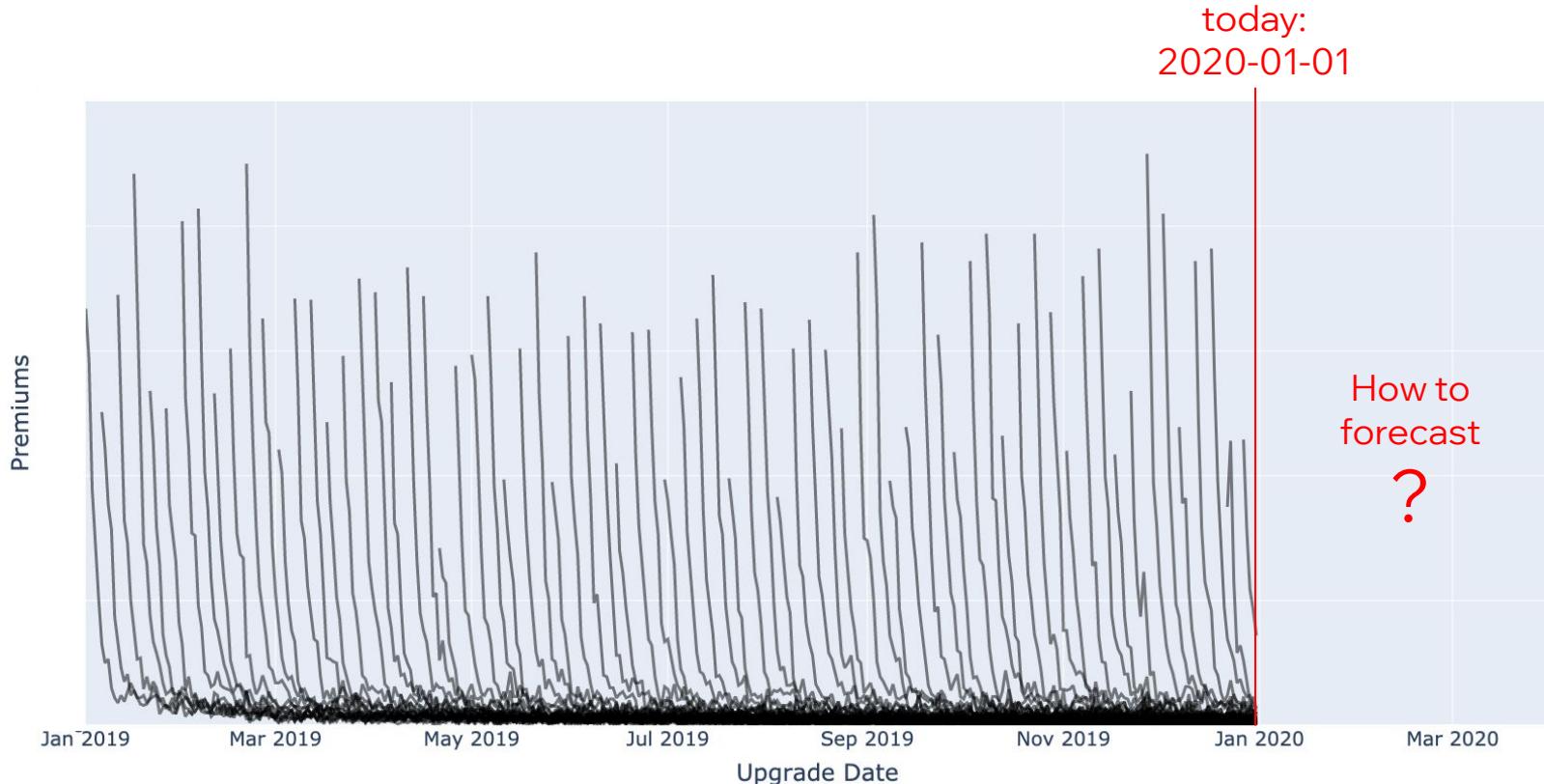
Recent cohorts registered during last year

today:
2020-01-01



Forecasting one quarter ahead

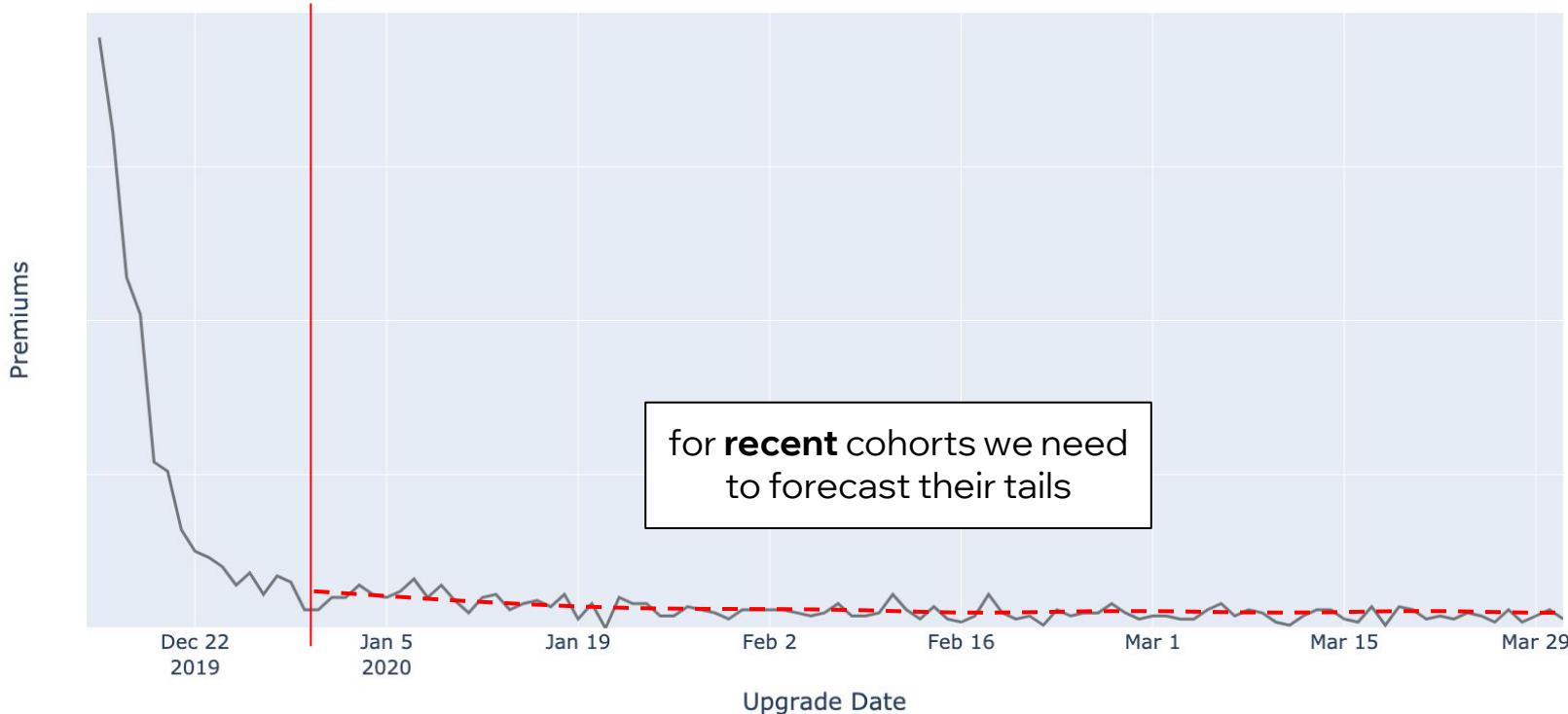
How to forecast?



Recent cohorts

forecast tails one quarter ahead

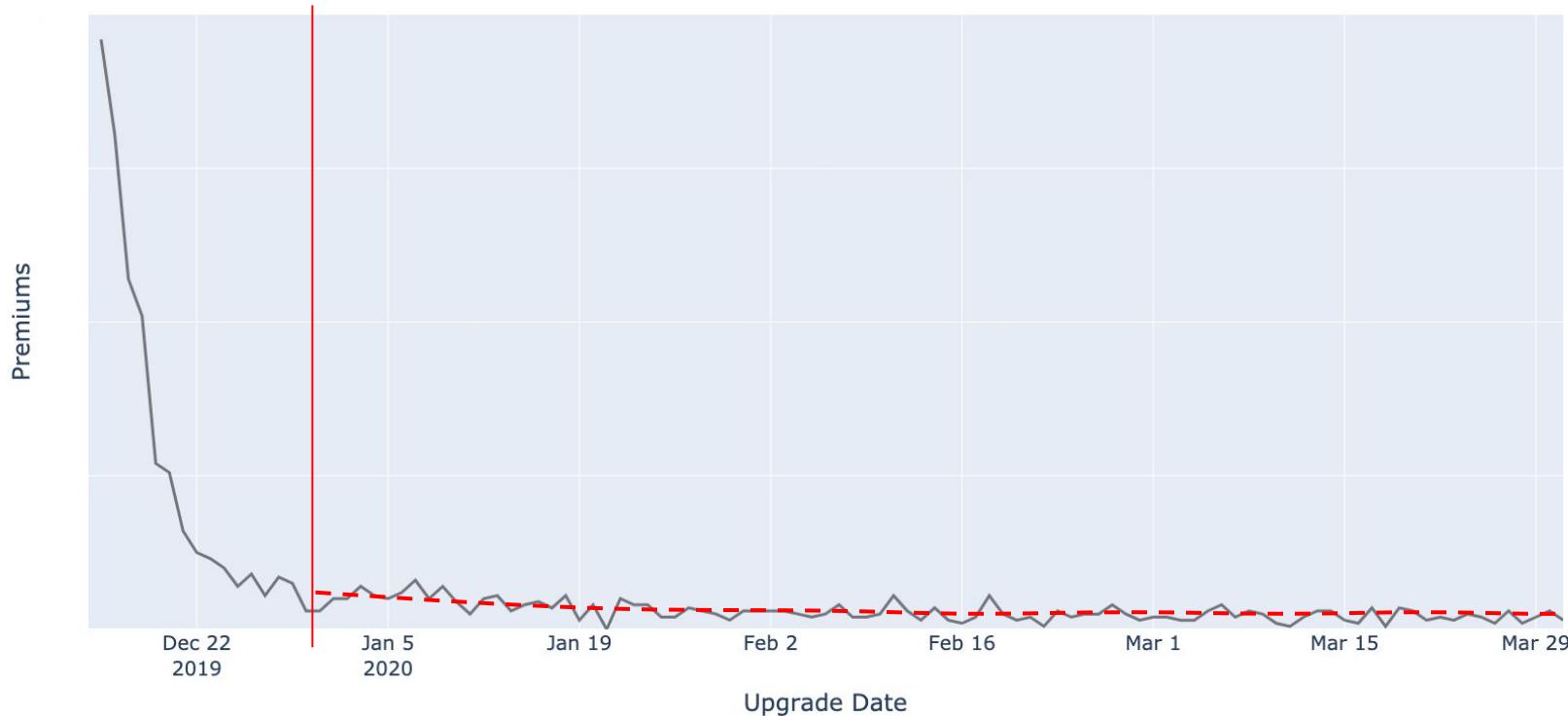
today:
2020-01-01



Recent cohorts

forecast tails one quarter ahead

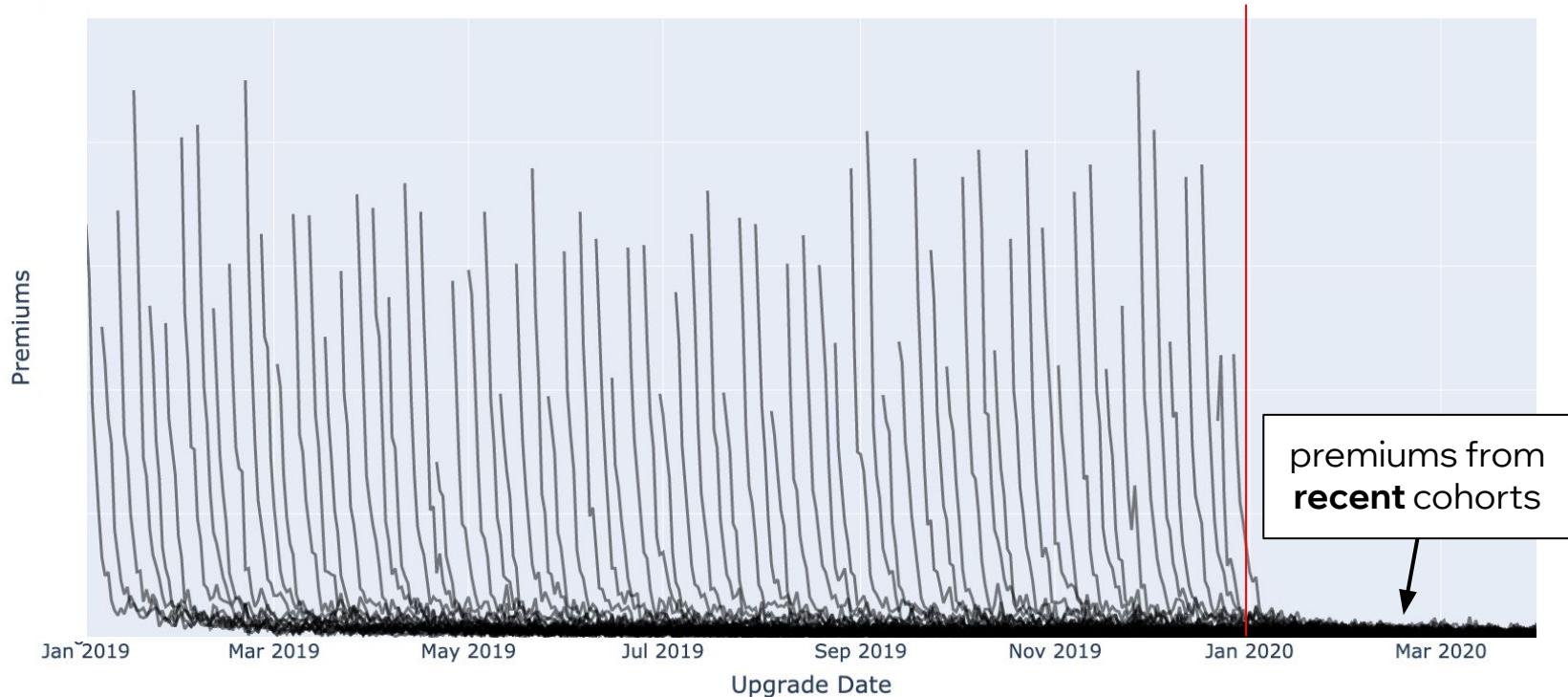
today:
2020-01-01



Recent cohorts

forecast tails one quarter ahead

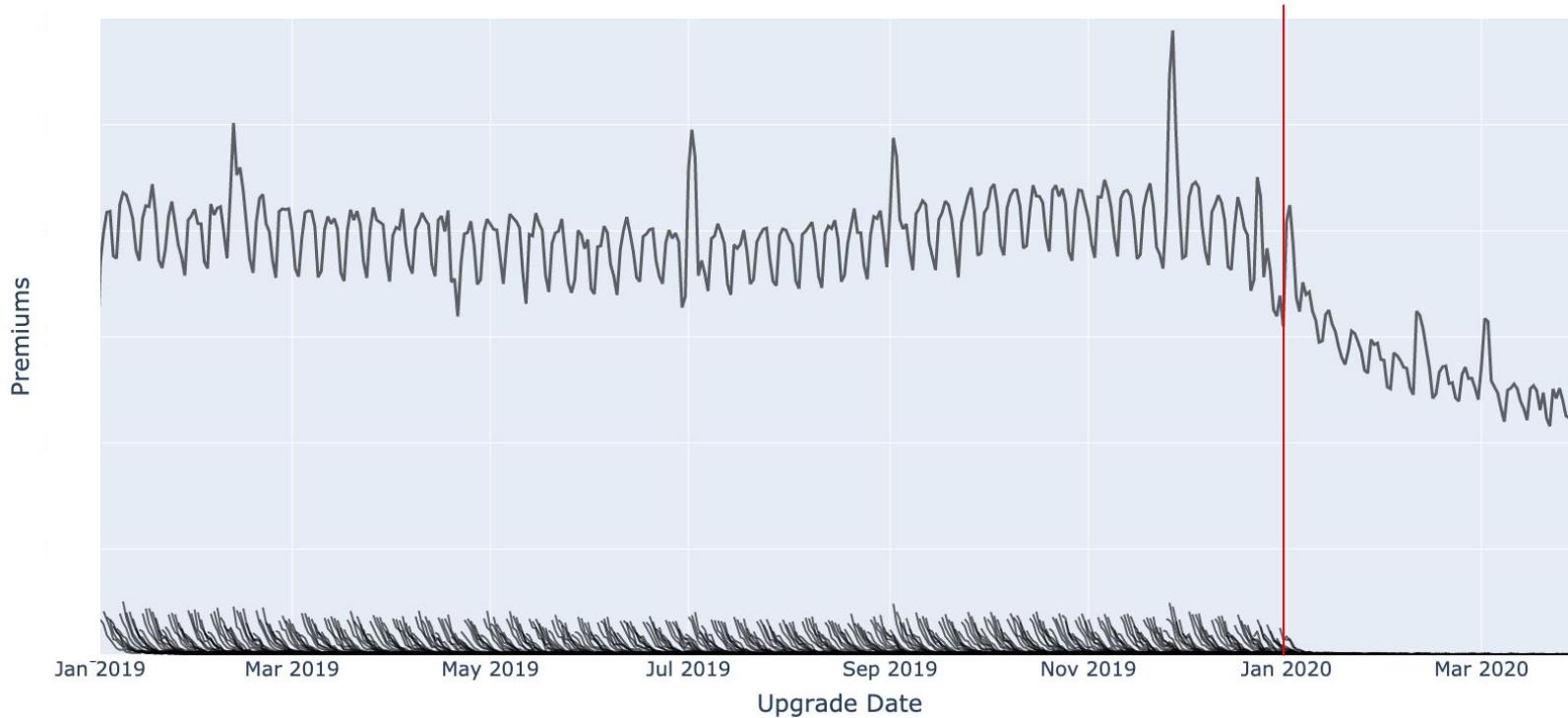
today:
2020-01-01



Recent cohorts

aggregated by upgrade date

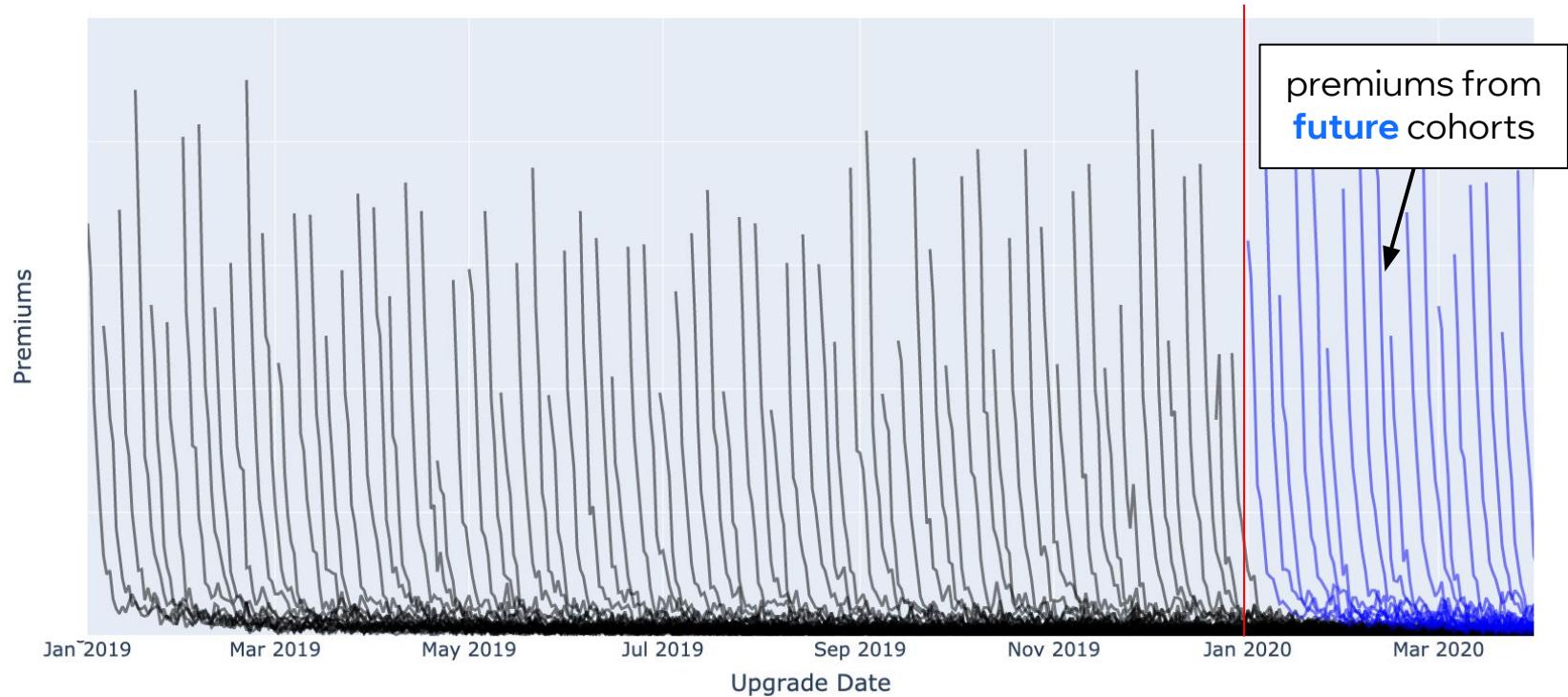
today:
2020-01-01



Future cohorts

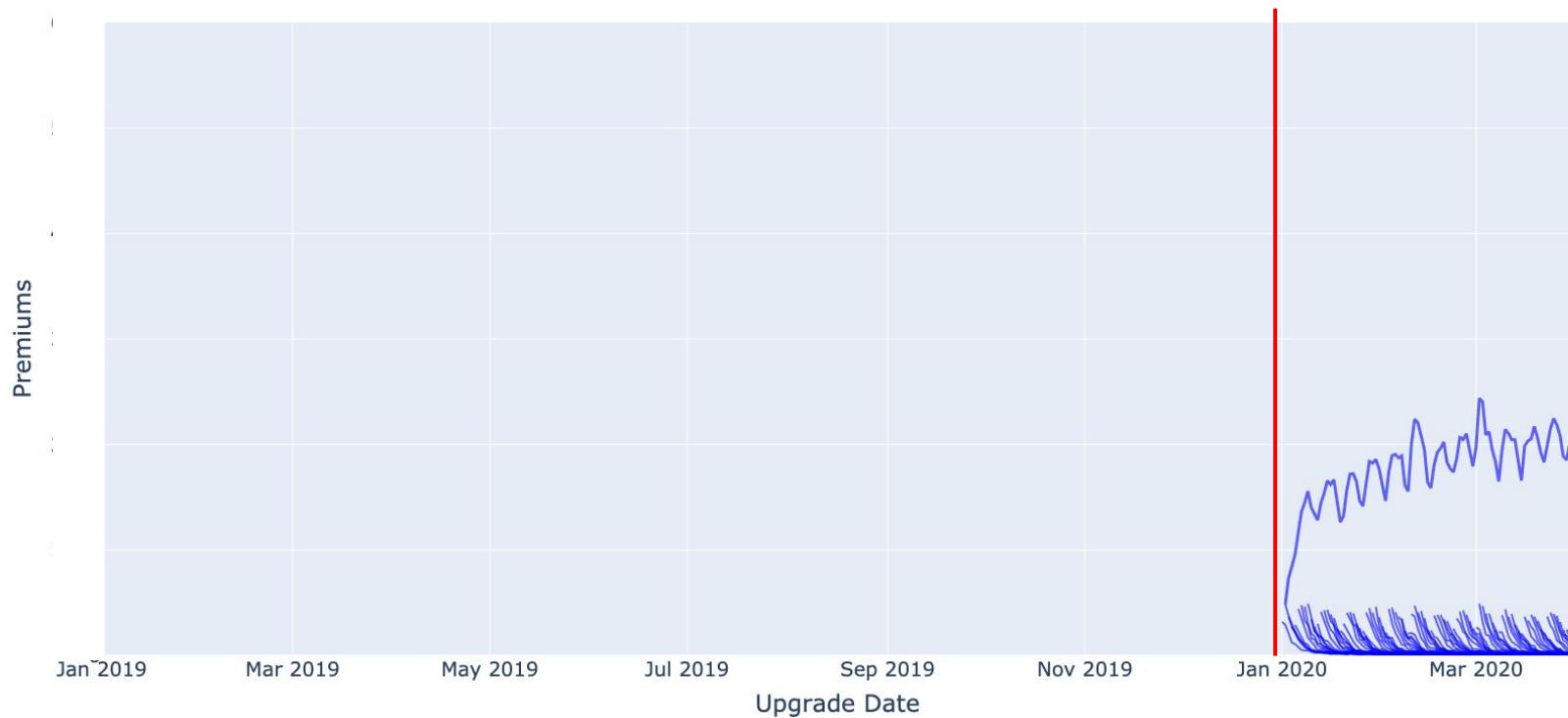
registered after train date, during the forecast quarter

today:
2020-01-01



Future cohorts

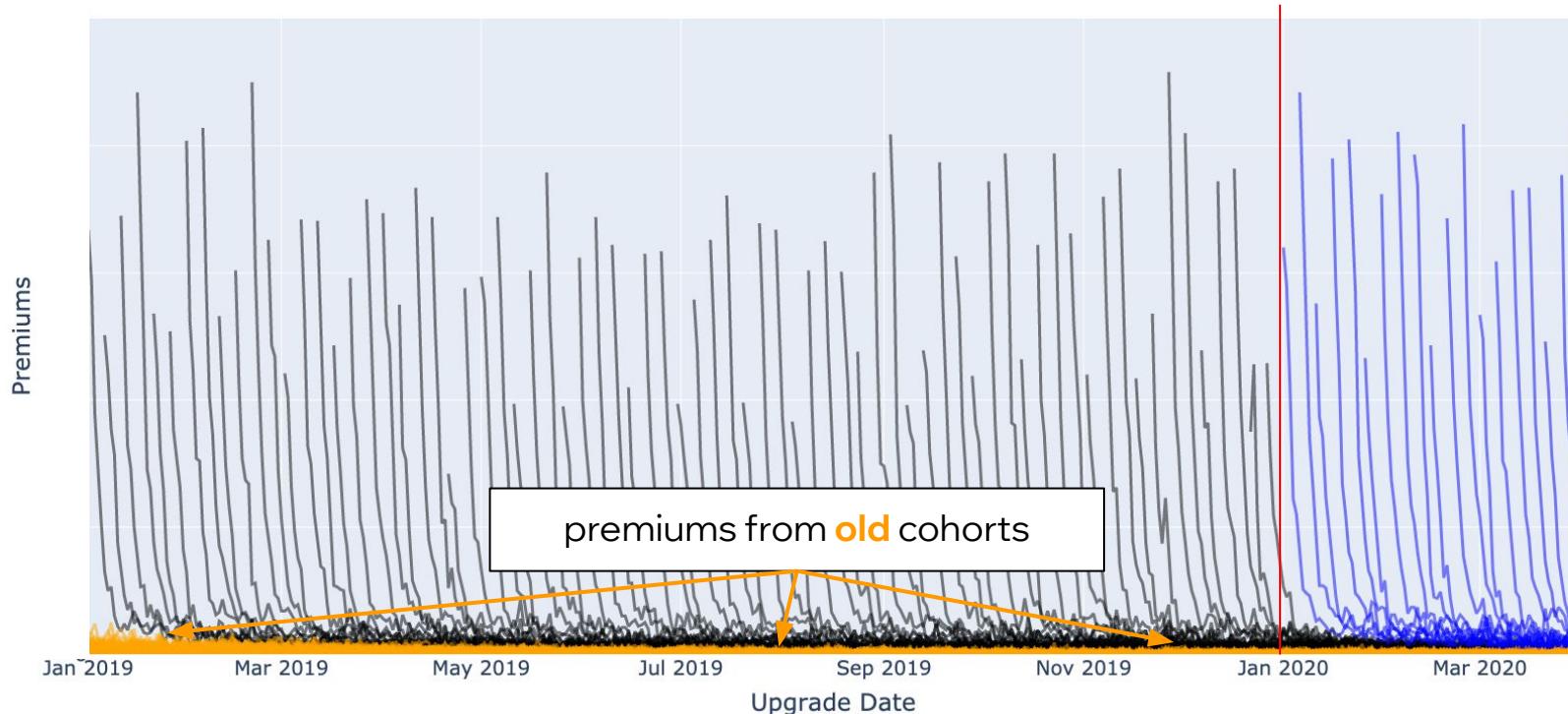
aggregated by upgrade date



Old cohorts

generate premiums long time after registration

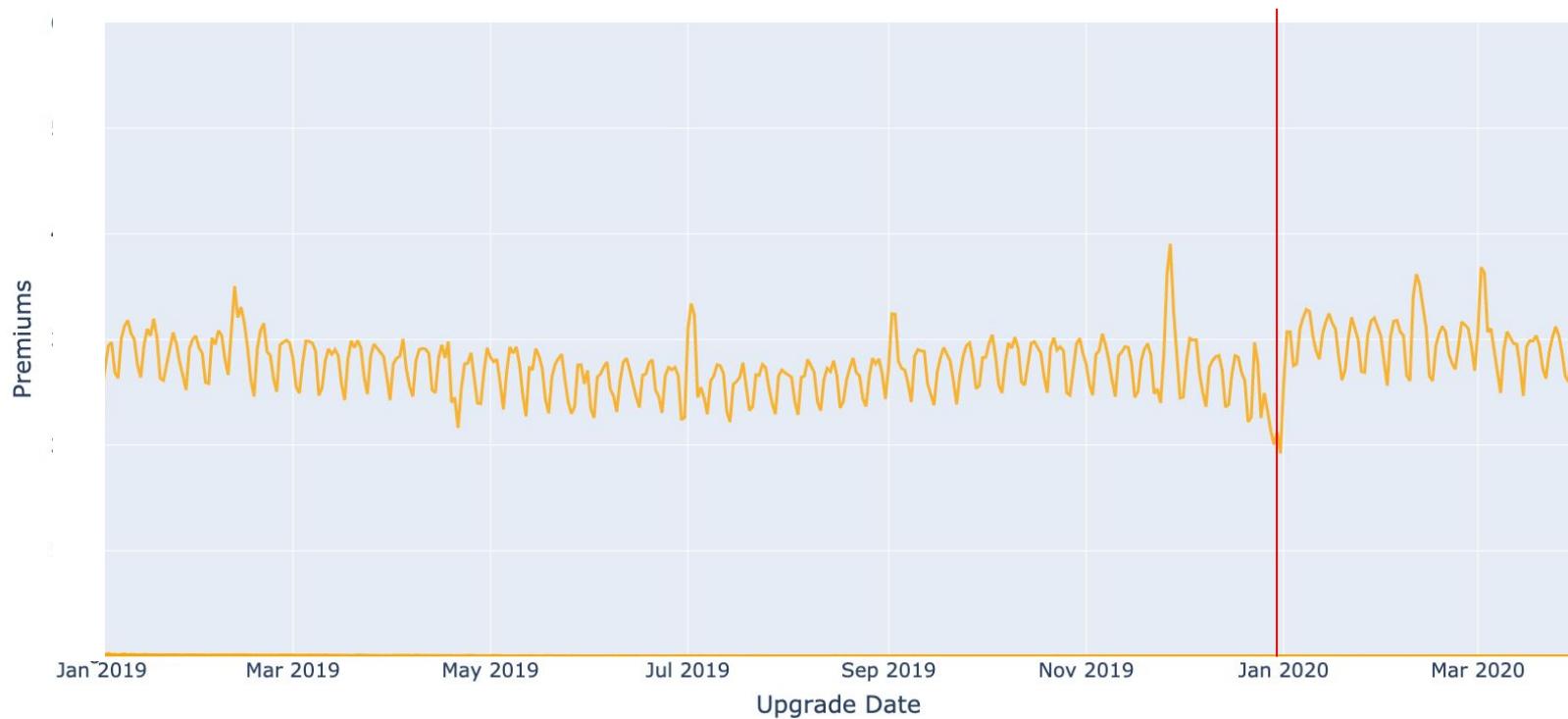
today:
2020-01-01



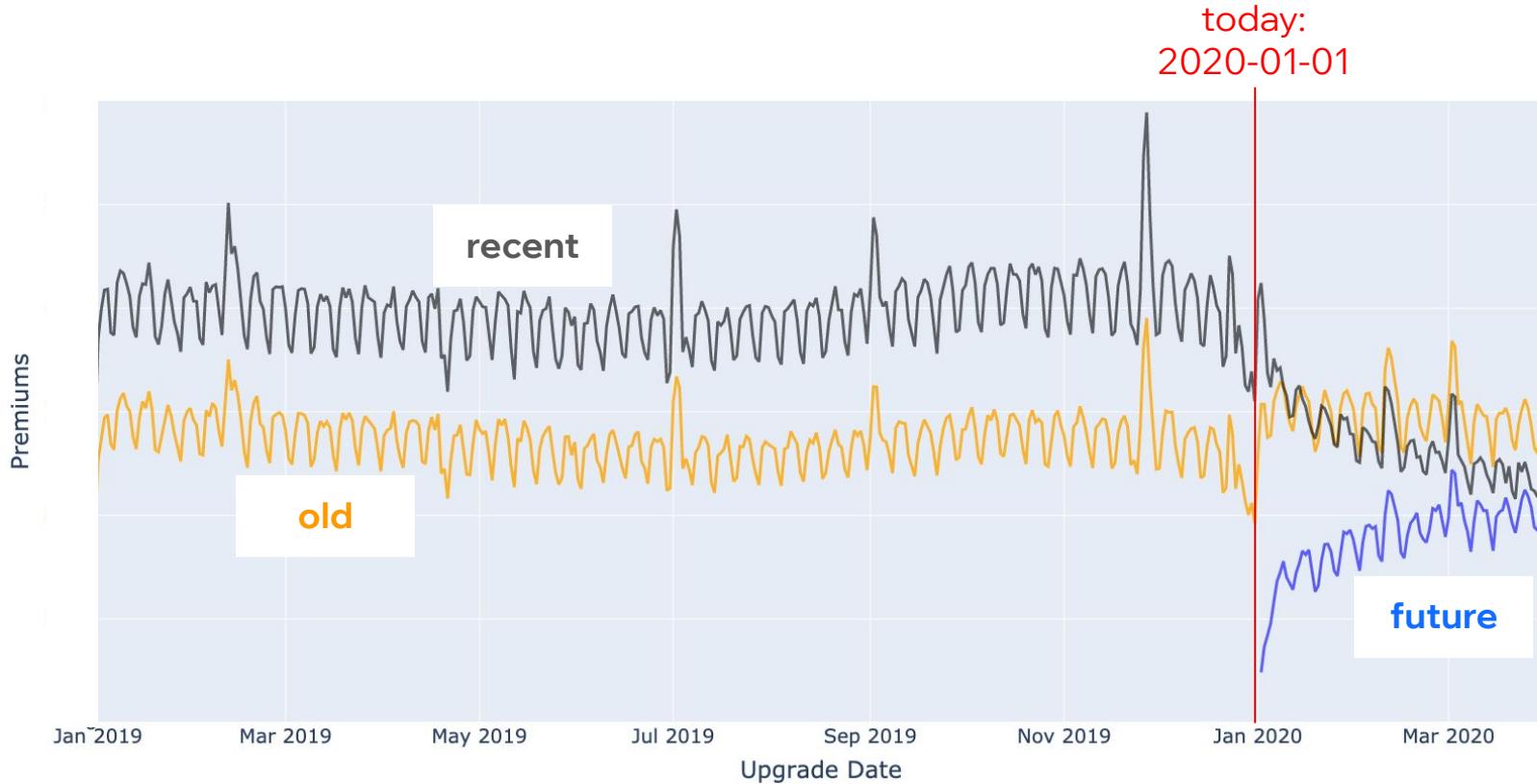
Old cohorts

aggregated by upgrade date

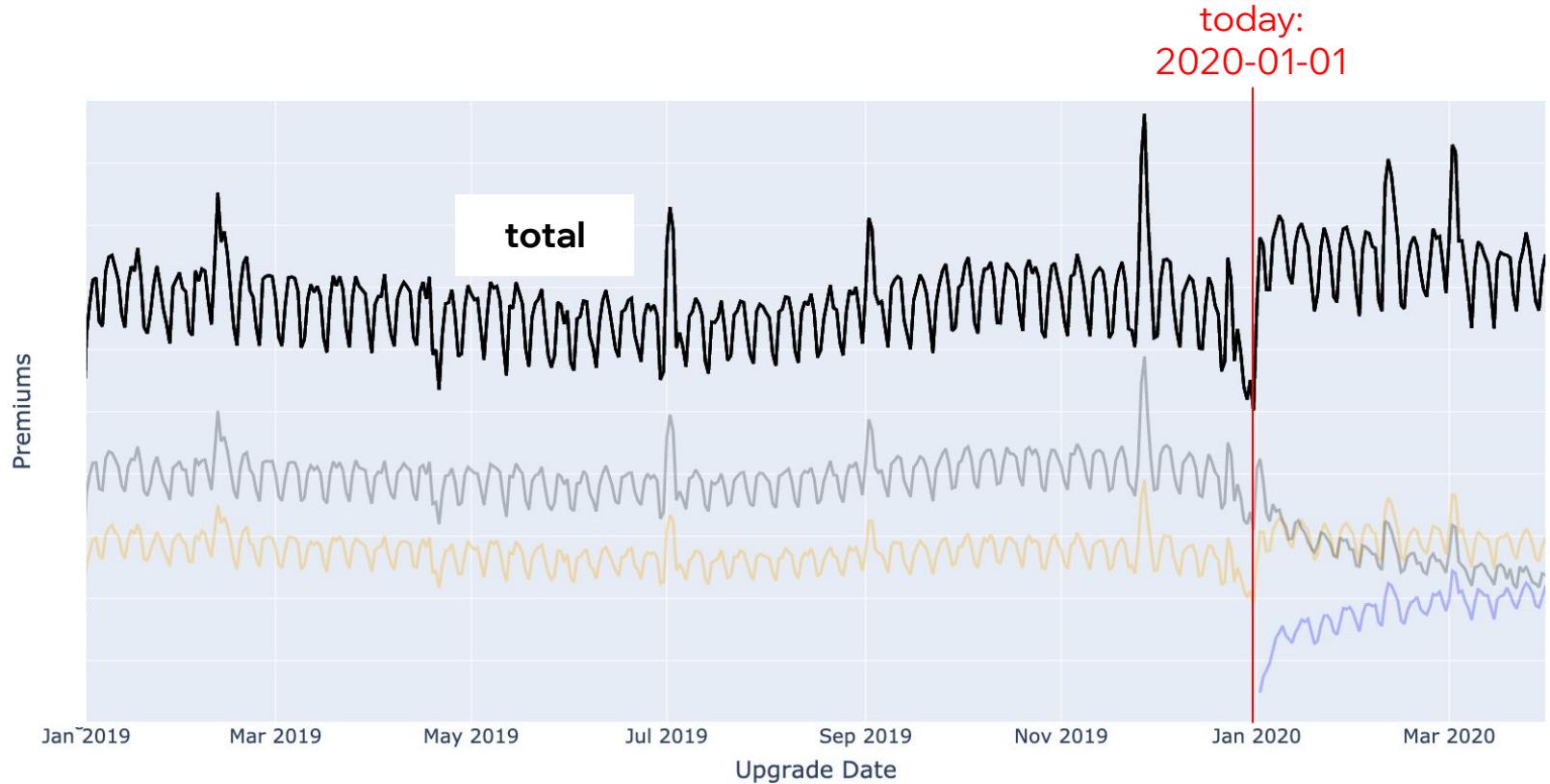
today:
2020-01-01



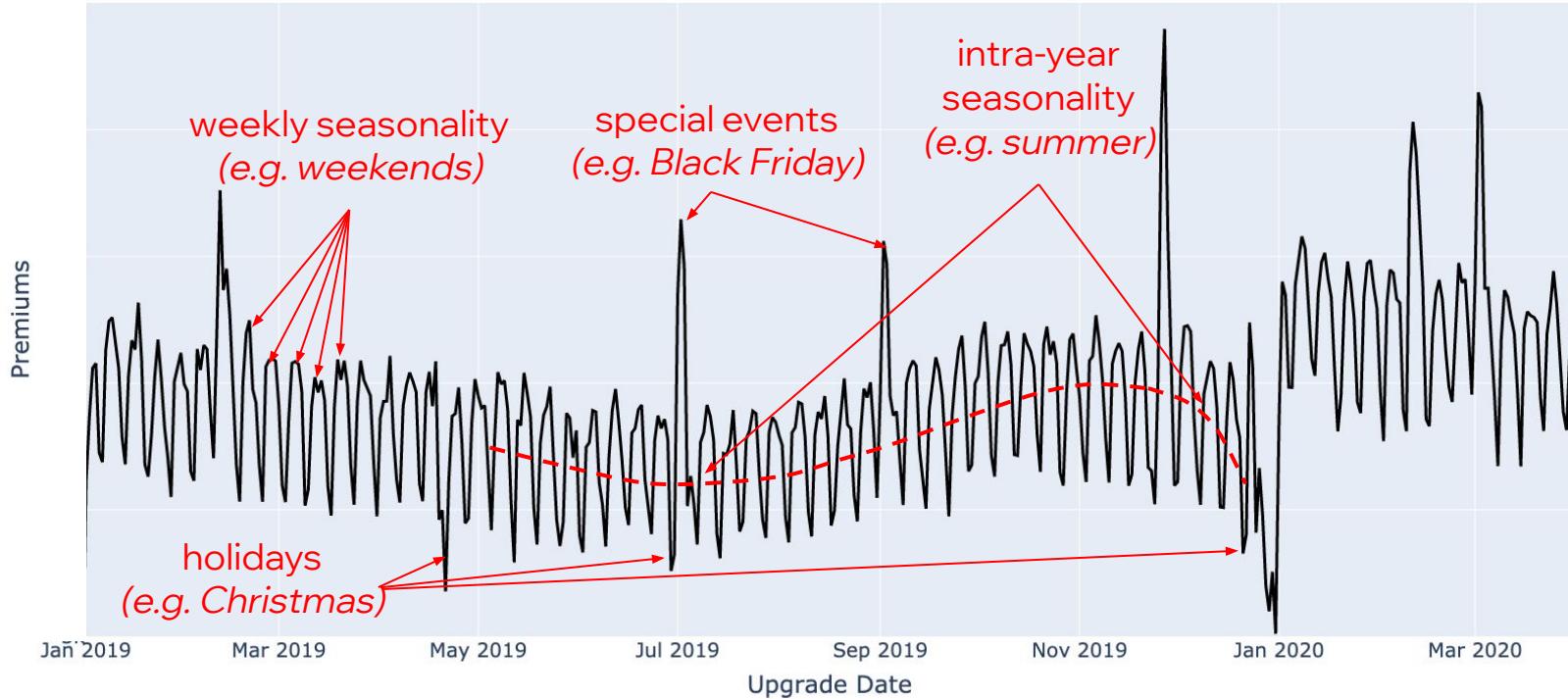
Old, Recent and Future cohorts comparison



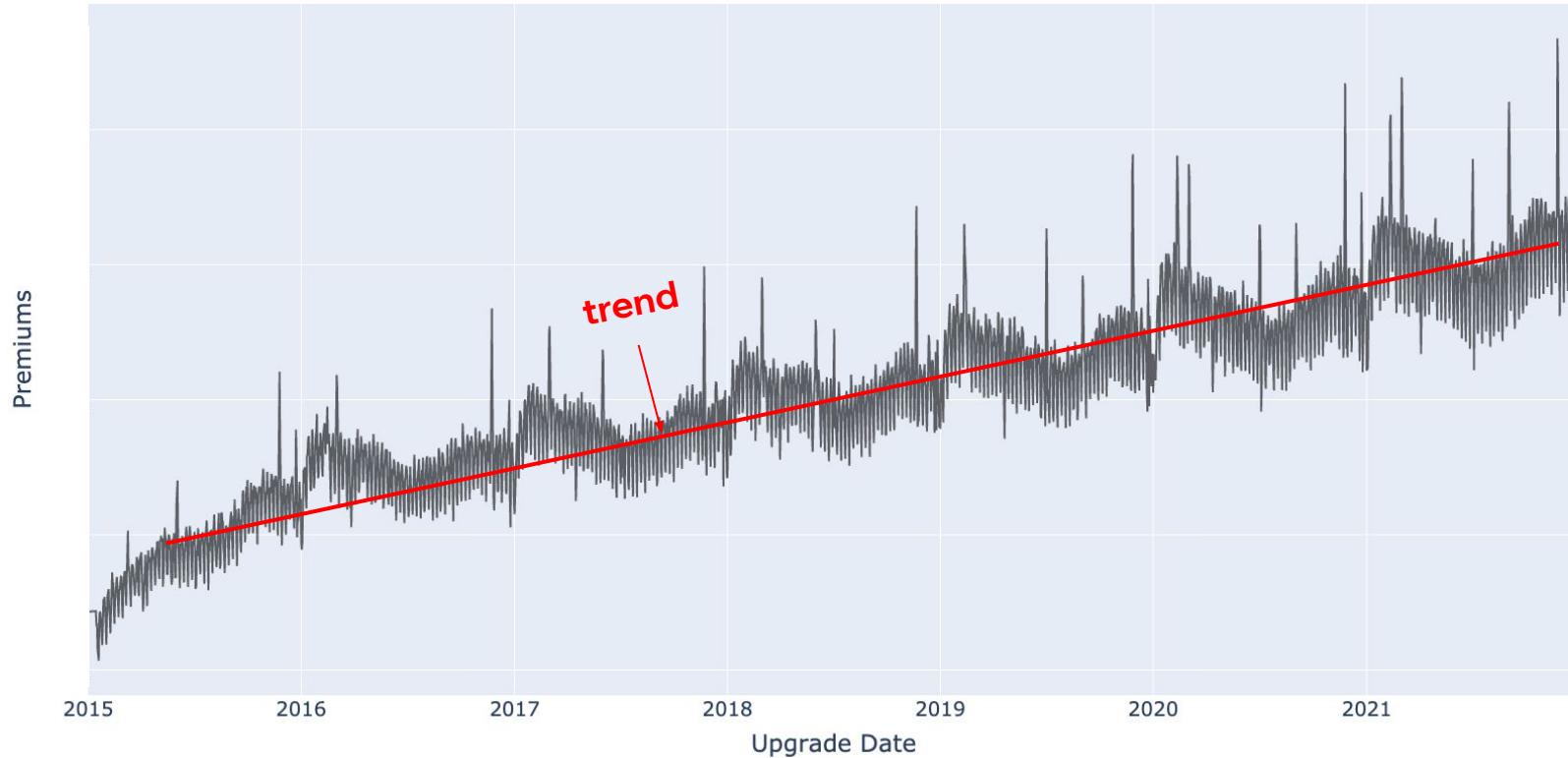
Old, Recent and Future cohorts add up to a time-series of total premiums



Time series of premiums has some interesting properties



Time series of premiums has some interesting properties



Using the setup for a **cohort-based model** we transform a **time-series** task into a **regression** task.

Regression models for cohort-based forecasting

Time-series data and models

date-value pairs + exogenous variables

time-series (date-value pairs)		exogenous variables	
date	premiums	events	holiday
2019-01-01	234	0	1
2019-01-02	456	0	0
2019-01-03	462	0	0
2019-01-04	1345	1	0
2019-01-05	1017	1	0
2019-01-06	962	1	0
2019-01-07	531	0	0
2019-01-08	385	0	0
today
2020-01-01	?	0	1
2020-01-02	?	0	0

*synthetic data

Table data for cohorts

values by double key: registration & upgrade date (age)

	double key		target		features		
	registration date	upgrade date	premiums	age	events	holiday	s(weekday)
cohort of reg day 2019-01-01	2019-01-01	2019-01-01	200	1	0	1	0.13
	2019-01-01	2019-01-02	138	2	1	0	0.22

	2019-01-01	2020-03-26	2	450	0	0	0.05
cohort of reg day 2019-01-02	2019-01-02	2019-01-02	198	1	1	0	0.22
	2019-01-02	2019-01-03	125	2	1	0	0.34

	2020-01-01	2020-01-01	?	1	0	0	0.26
	2020-01-01	2020-01-02	?	2	0	0	0.09

	2020-01-01	2021-03-31	?	455	1	0	0.56

today

Cartesian product of registration dates and upgrade dates (age)
⇒ many rows produced

*synthetic data

Table data for cohorts

values by double key: registration & upgrade date (age)

	double key		target		features		
	registration date	upgrade date	premiums	age	events	holiday	s(weekday)
cohort of reg day 2019-01-01	2019-01-01	2019-01-01	200	1	0	1	0.13
	2019-01-01	2019-01-02	138	2	1	0	0.22

	2019-01-01	2020-03-26	2	450	0	0	0.05
cohort of reg day 2019-01-02	2019-01-02	2019-01-02	198	1	1	0	0.22
	2019-01-02	2019-01-03	125	2	1	0	0.34

	2020-01-01	2020-01-01	?	1	0	0	0.26
	2020-01-01	2020-01-02	?	2	0	0	0.09

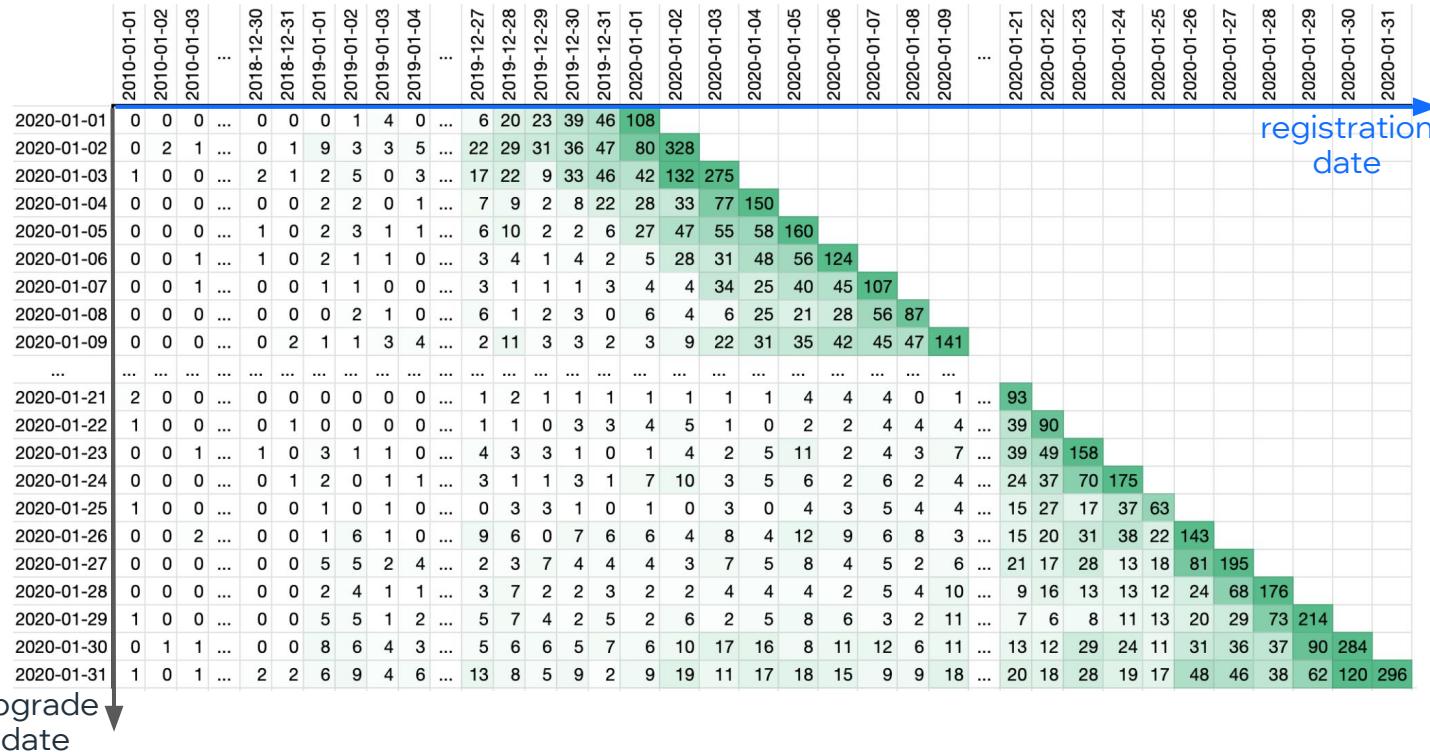
	2020-01-01	2021-03-31	?	455	1	0	0.56

today

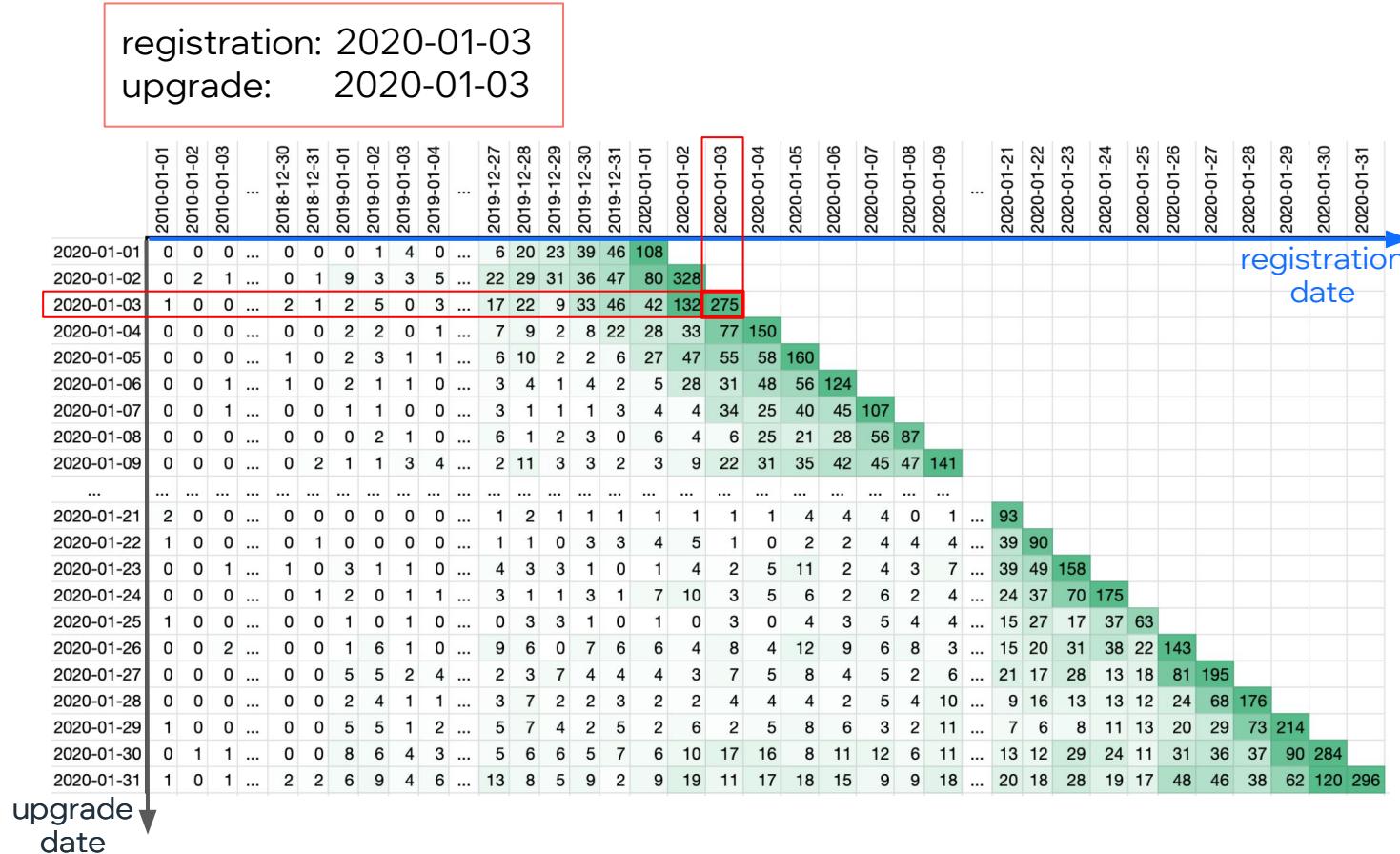
Cartesian product of registration dates and upgrade dates (age)
⇒ many rows produced

*synthetic data

Cohorts in triangle by registration and upgrade date

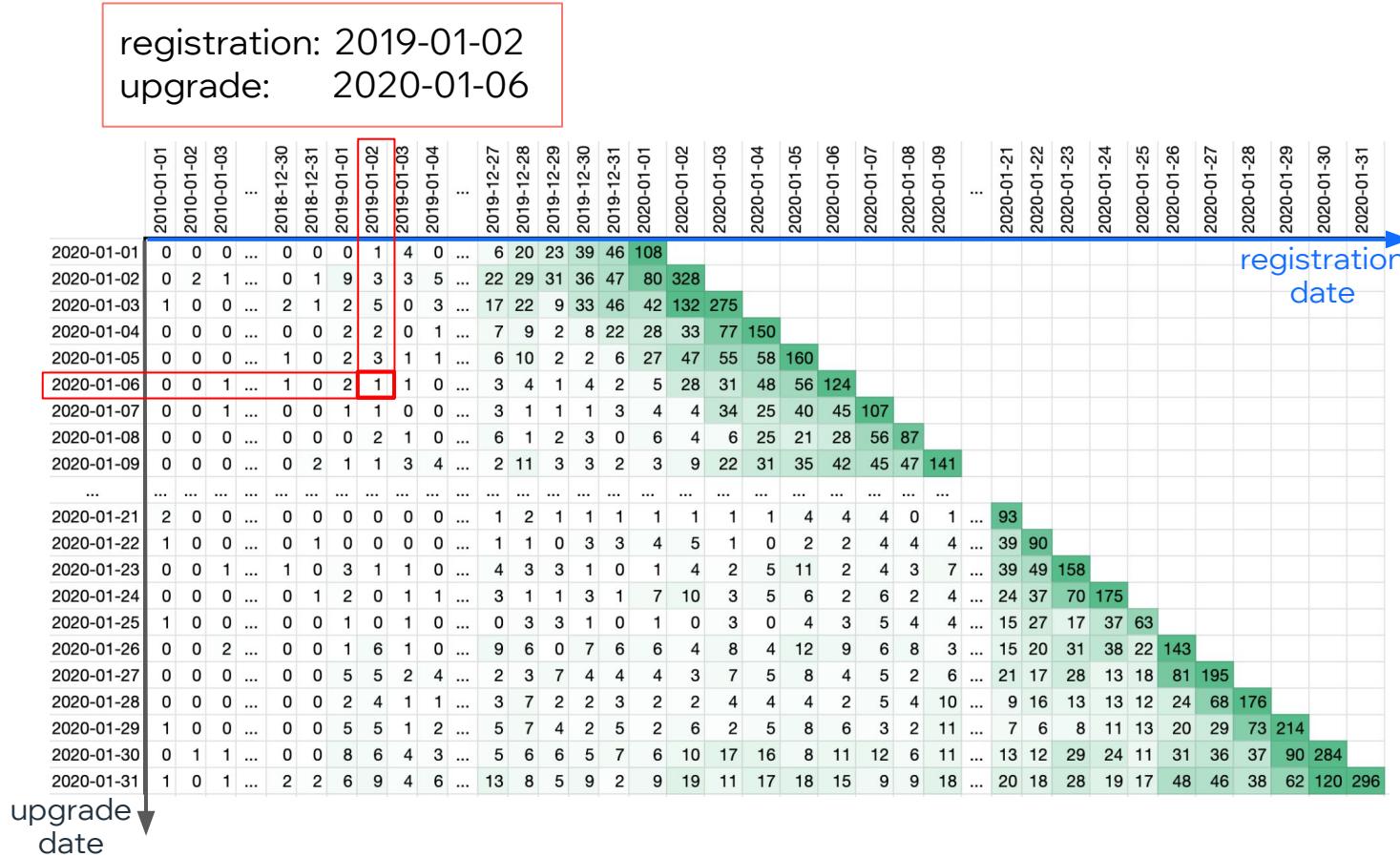


Cohorts in triangle by registration and upgrade date



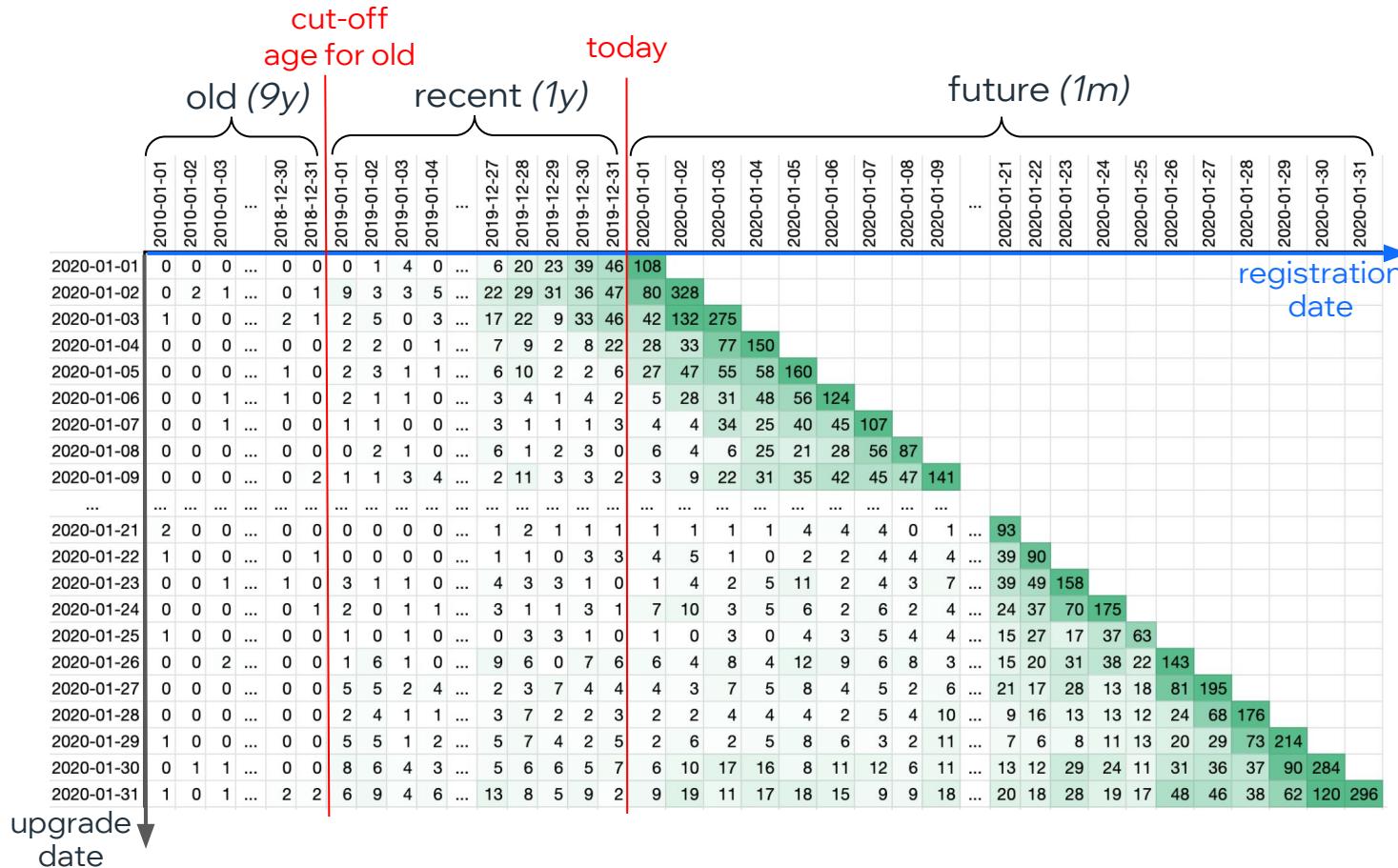
*synthetic data

Cohorts in triangle by registration and upgrade date



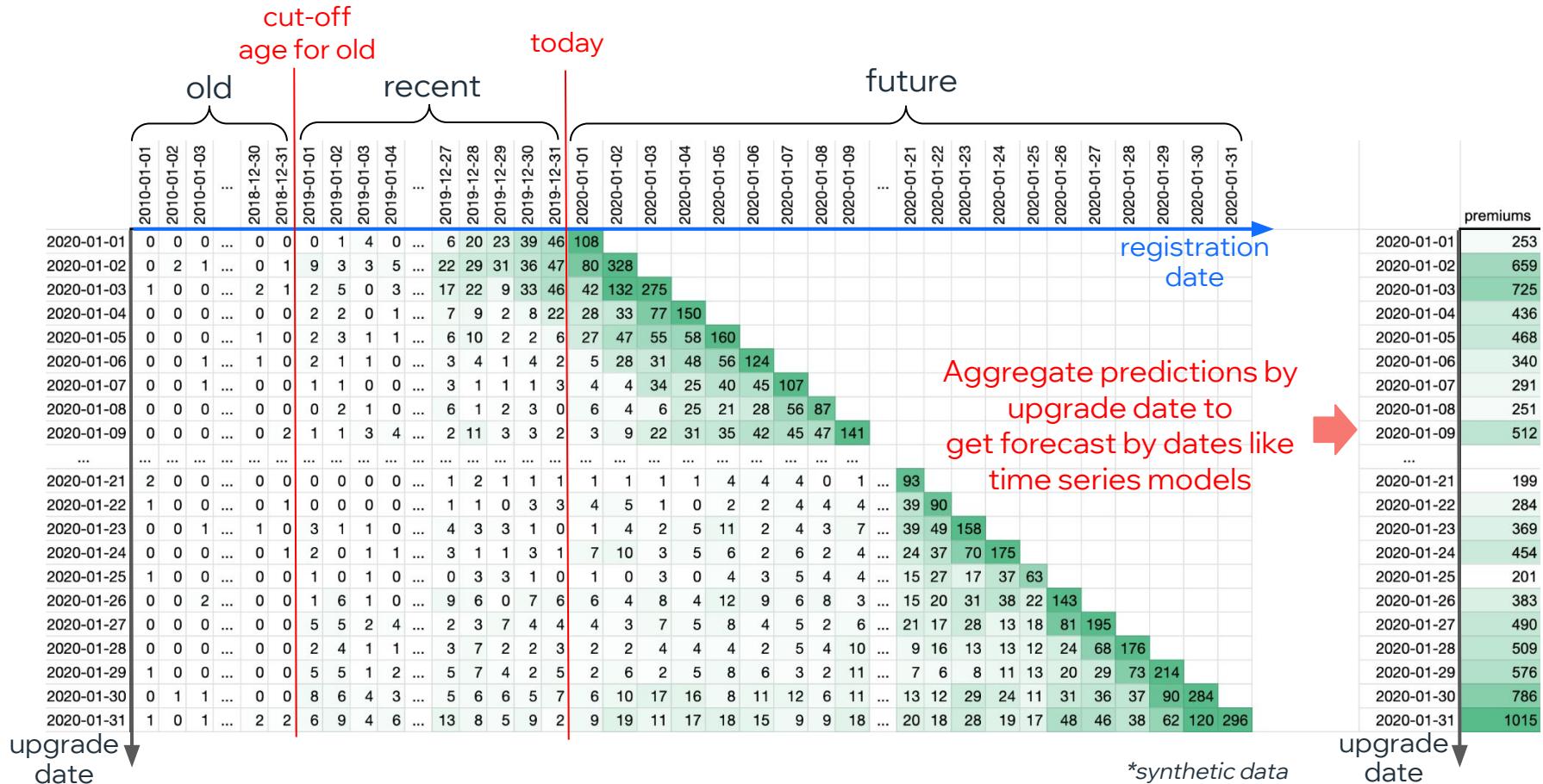
*synthetic data

Cohorts in triangle by registration and upgrade date



*synthetic data

Cohorts in triangle by registration and upgrade date



Regression models

Generalized Linear Model (GLM)

link
function

linear
predictor

probability
distribution

$$g(\mu) = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$$

where: $y \sim P(\mu)$

$$E(y | X) = \mu = g^{-1}(X\beta)$$

- Non-normal distribution of y (e.g. from exponential family)
- Linearity between transformed response in terms of the link function and regressors
- Homoscedasticity NOT necessary
- Errors independent, but NOT necessarily normal
- Estimate with MLE instead of OLS

Distribution	Link function
Normal	Identity
Binomial	Logit
Poisson	Log
Gamma	Negative inverse

Regression models

GLM with Poisson distribution and log link function

$\text{premiums}_{(t,r)} \sim \text{Poisson}(\lambda_{(t,r)})$

$$\begin{aligned}\ln(\lambda_{(t,r)}) = & \text{age}_{(t,r)} + \text{weekday}(t) + \text{dayofyear}(t) \\ & + \text{holiday}(t) + \text{special_events}_{(t,r)} + \dots\end{aligned}$$

where: **t** - upgrade date; **r** - registration date

Pros:

- Supports non-normal distributions from the exponential family (Poisson, Gamma, Tweedie)
- Can extrapolate the trend well

Cons:

- Needs manual feature engineering
- Few python packages have all distributions and regularization
- sensitive to the choice of distribution and link func

Regression models

GLM using statsmodels library in Python

```
from statsmodels.genmod.generalized_linear_model import GLM
from statsmodels.genmod.families import Poisson

formula = 'premiums ~
            log1p_age + f1_age + ... + f10_age +
            f1_numdate + ... + f6_numdate
            f1_weekday + ... + f5_weekday +
            f1_dayofyear + ... + f10_dayofyear +
            holiday +
            f1_economy_change + ... + f1_fx_rate +
            special_event + f1_special_event_days + f2_special_event_days +
            new_version + package_update +
            log1p_age*new_version + log1p_age*special_event'

model = GLM.from_formula(formula=formula, data=df_train, family=Poisson())
model_fit = model.fit()
```

Regression models

GLM using *statsmodels* library in Python

```
from statsmodels.genmod.generalized_linear_model import GLM
from statsmodels.genmod.families import Poisson

formula = 'premiums ~
           decay by age {log1p_age + f1_age + ... + f10_age +
           time-series decomp: {f1_numdate + ... + f6_numdate
           trend, seasonality, holidays {f1_weekday + ... + f5_weekday +
           f1_dayofyear + ... + f10_dayofyear +
           holiday +
           external factors {f1_economy_change + ... + f1_fx_rate +
           business {special_event + f1_special_event_days + f2_special_event_days +
           new_version + package_update +
           interaction terms {log1p_age*new_version + log1p_age*special_event'}
```

```
model = GLM.from_formula(formula=formula, data=df_train, family=Poisson())
model_fit = model.fit()
```

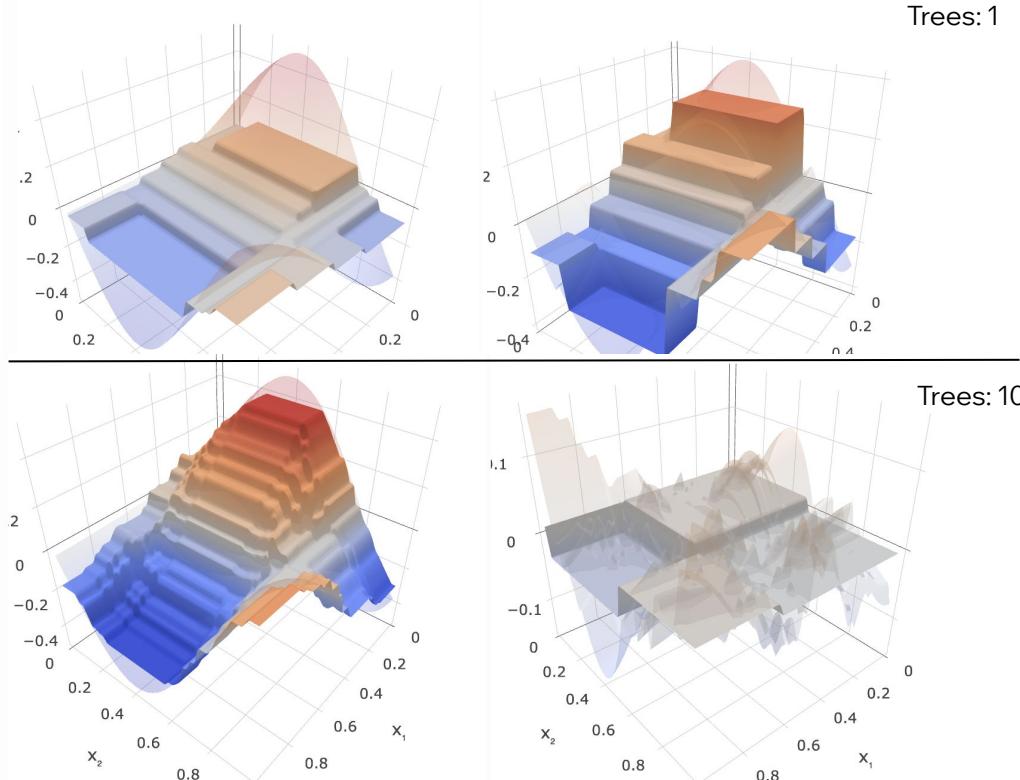
Note: **f1_, f2_...** prefixes mean that this feature is a transformation of the raw feature, for example with a spline function

Regression models

Gradient Boosted Machine

target function $f(\mathbf{x})$ and prediction of previous trees $D(\mathbf{x})$

residual $R(\mathbf{x})$ and prediction of next tree $d_n(\mathbf{x})$



How it works:

- Each tree is trained on residuals from the previous iteration, taken with a small rate
- Prediction is reconstructed from the residuals predicted by all trees

blog: [**Brilliantly wrong**](#)

* pictures from: <https://arogozhnikov.github.io/>

Regression models

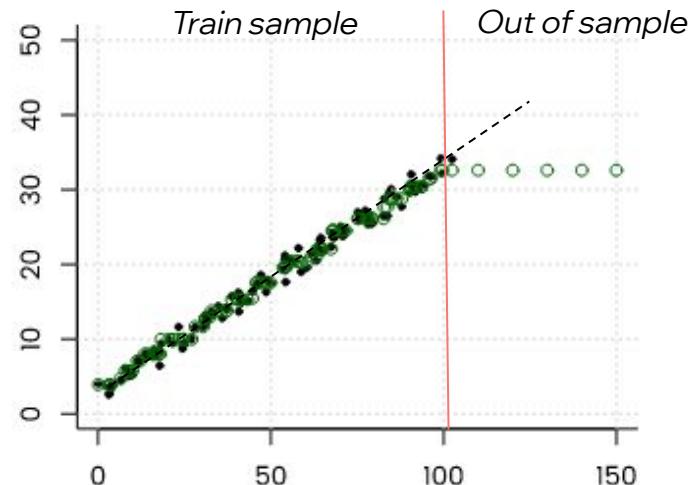
Gradient Boosted Machine

Pros:

- Handles non-linearities and interactions, none or little feature engineering is needed
- Supports non-normal distributions from the exponential family (Poisson, Gamma, Tweedie)
- Less sensitive to the choice of distribution

Cons:

- Can't extrapolate on unseen data, e.g. can't extrapolate the trend.



Regression models

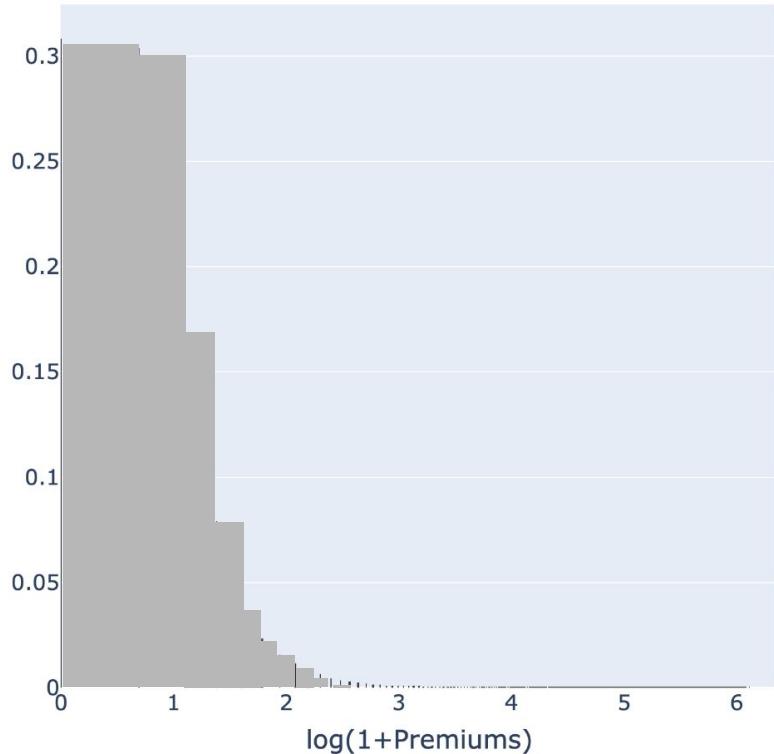
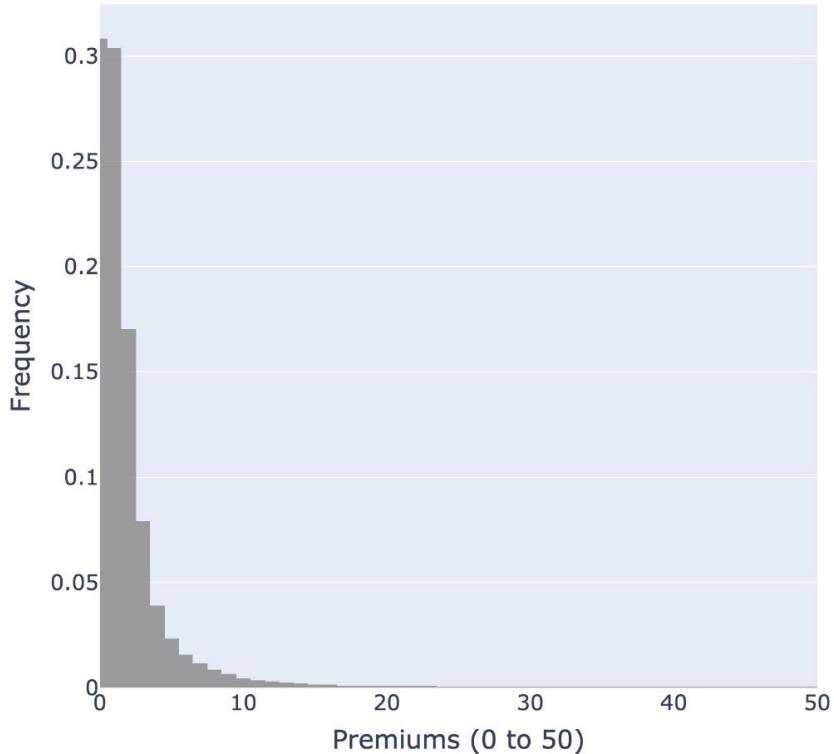
GBM using *lightgbm* in Python

```
import lightgbm
features = ['age', 'numdate', 'weekday', 'dayofyear', 'economy_change',
            'fx_rate', 'special_event', 'new_version', 'package_update']
params = {
    'objective': 'poisson',
    'boosting': 'gbdt',
    'max_depth': 4,
    'num_leaves': 15,
    'min_data_in_leaf': 200,
    'feature_fraction': 0.50,
    'bagging_fraction': 0.50,
    'learning_rate': 0.15
}
model = lightgbm.train(
    params=params, train_set=df_train, valid_sets=[df_valid, df_train],
    num_boost_round=1000,
    early_stopping_rounds=100
)
```

Few important recommendations

Choose the right distribution

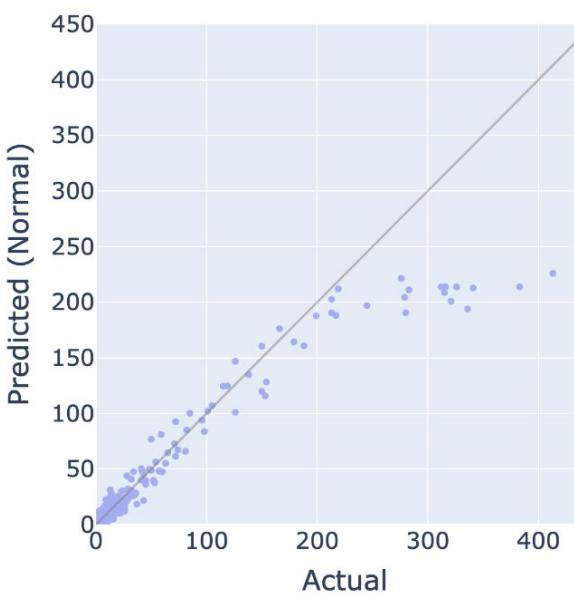
Target is non-normal, highly skewed, exponential-like



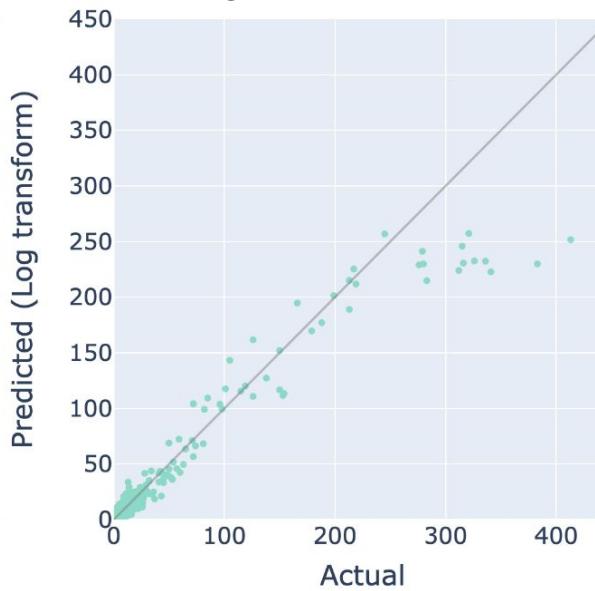
Choose the right distribution

Normal vs Log-transformed vs Poisson in GLM regression

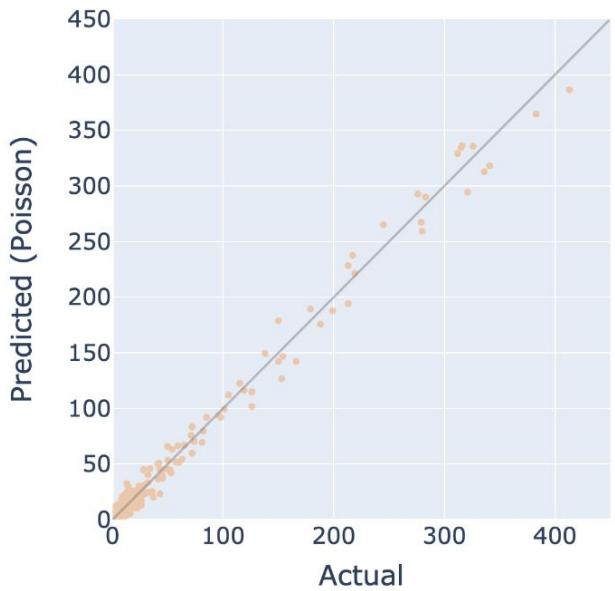
Normal



Normal,
log transformation

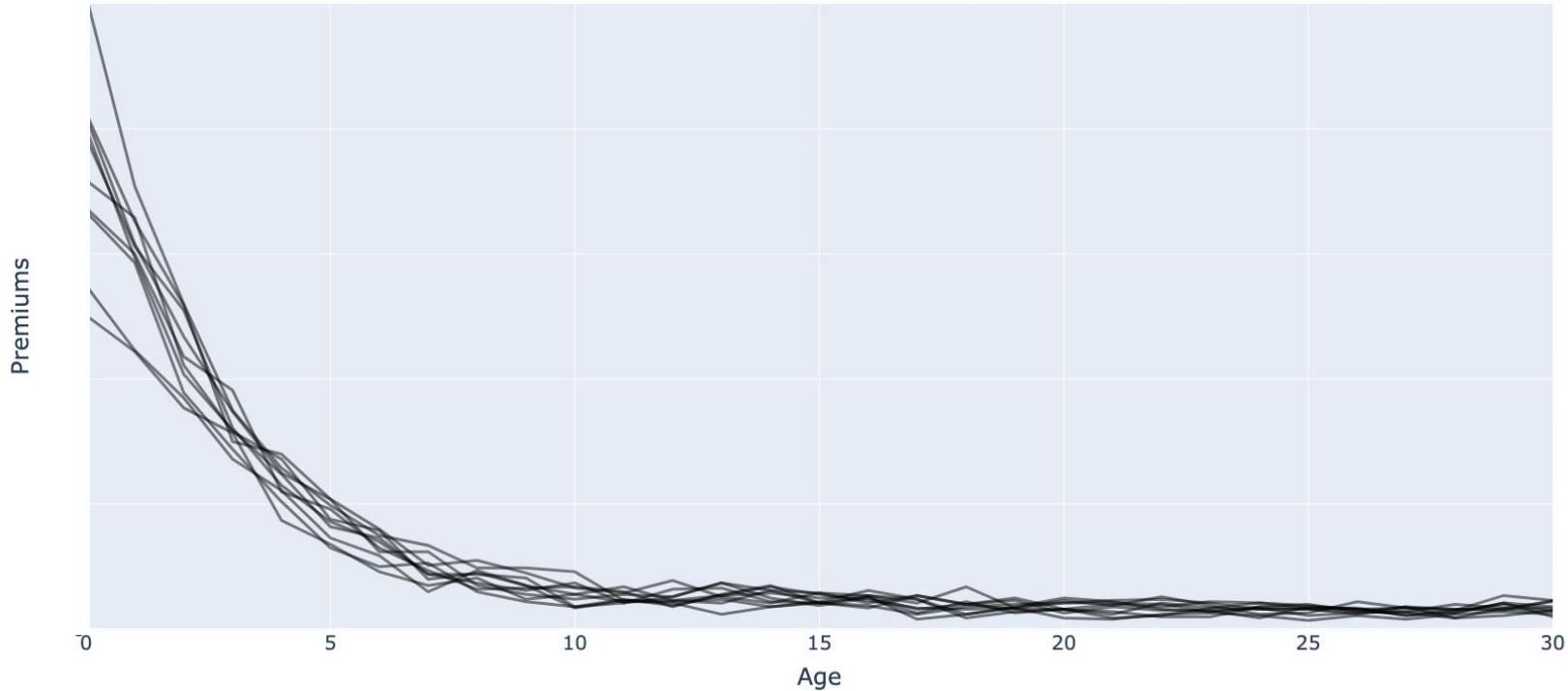


Poisson



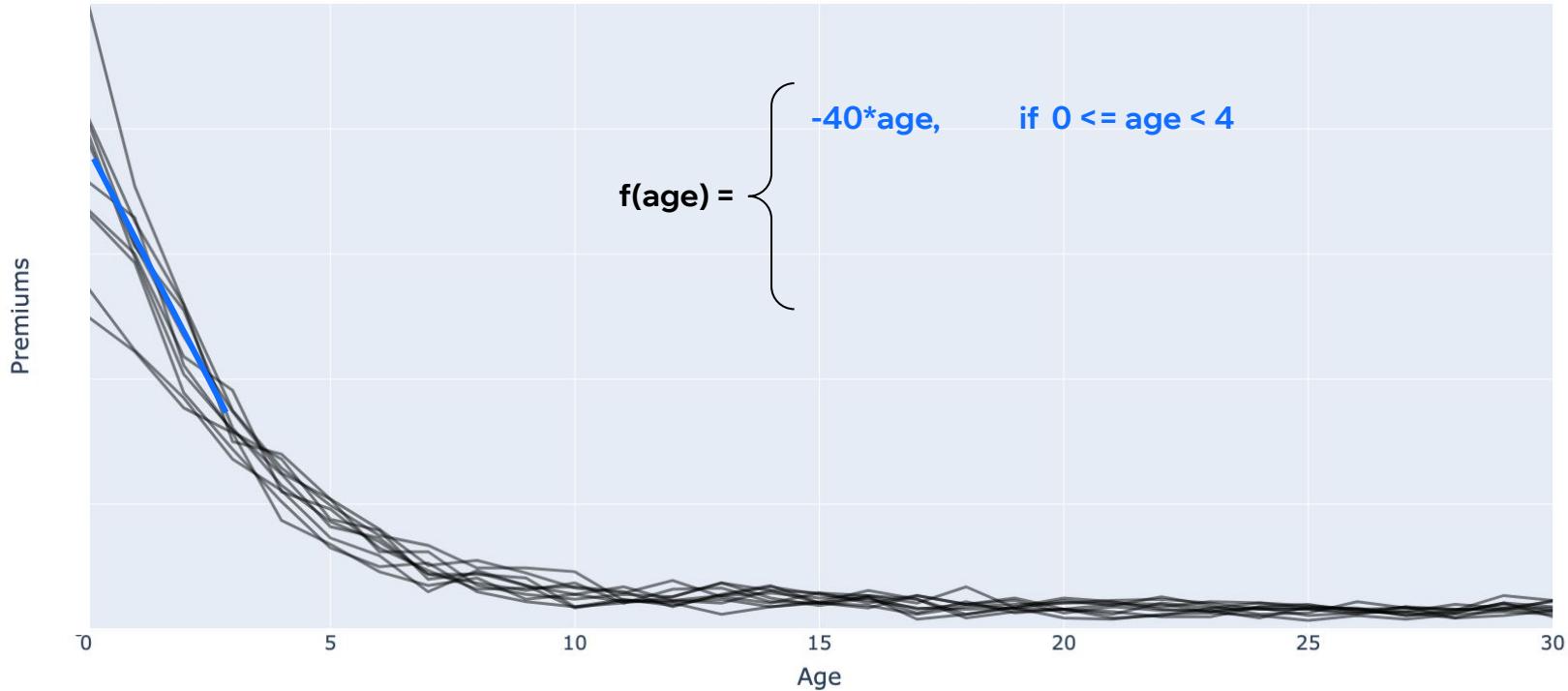
Feat engineering: decay of premiums by age

Piecewise linear function



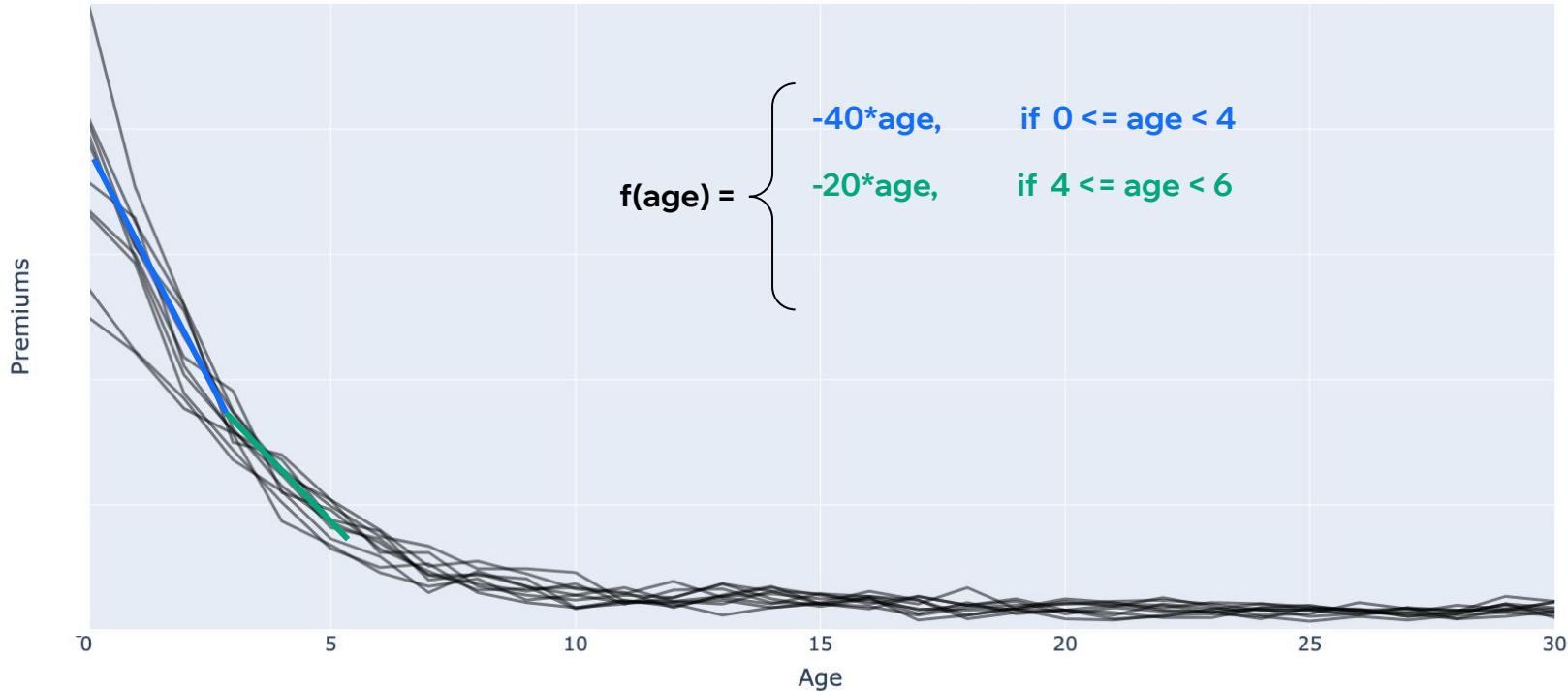
Feat engineering: decay of premiums by age

Piecewise linear function



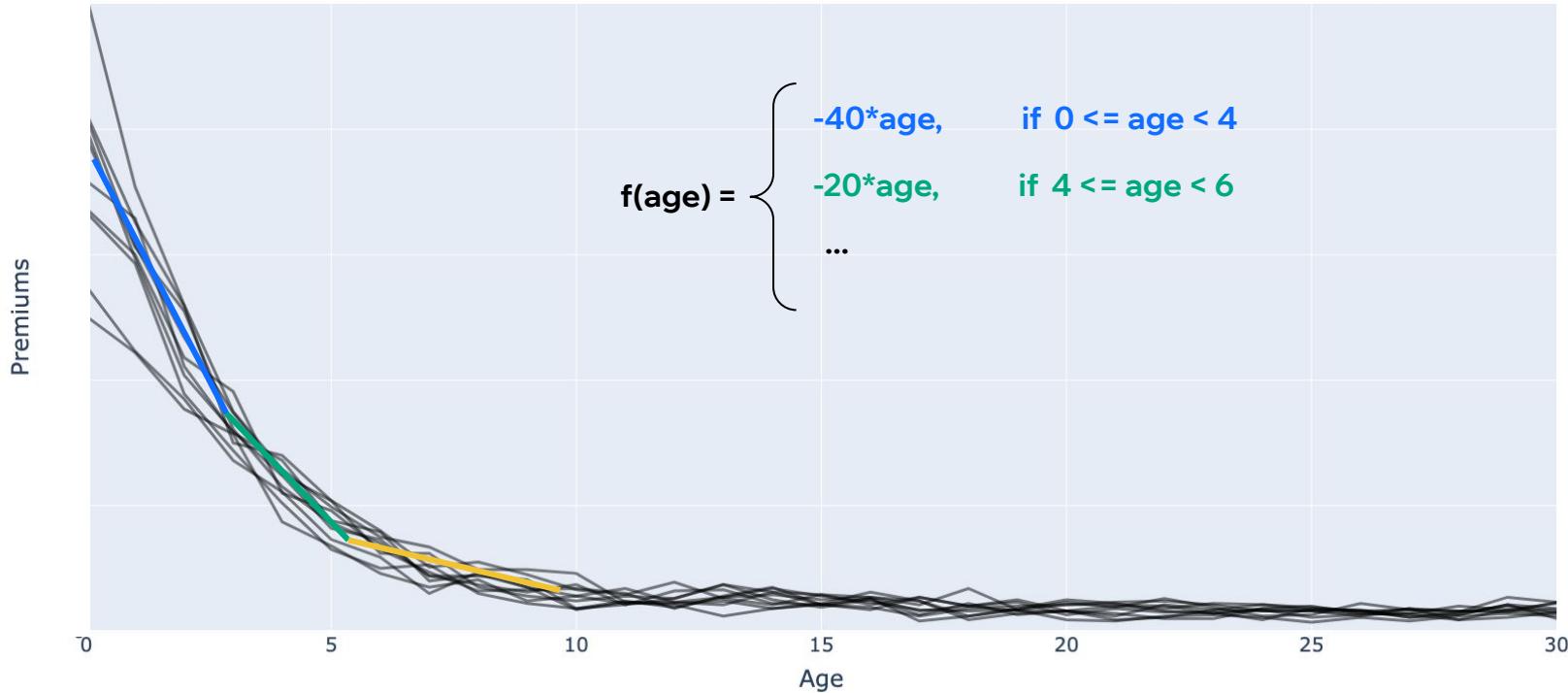
Feat engineering: decay of premiums by age

Piecewise linear function



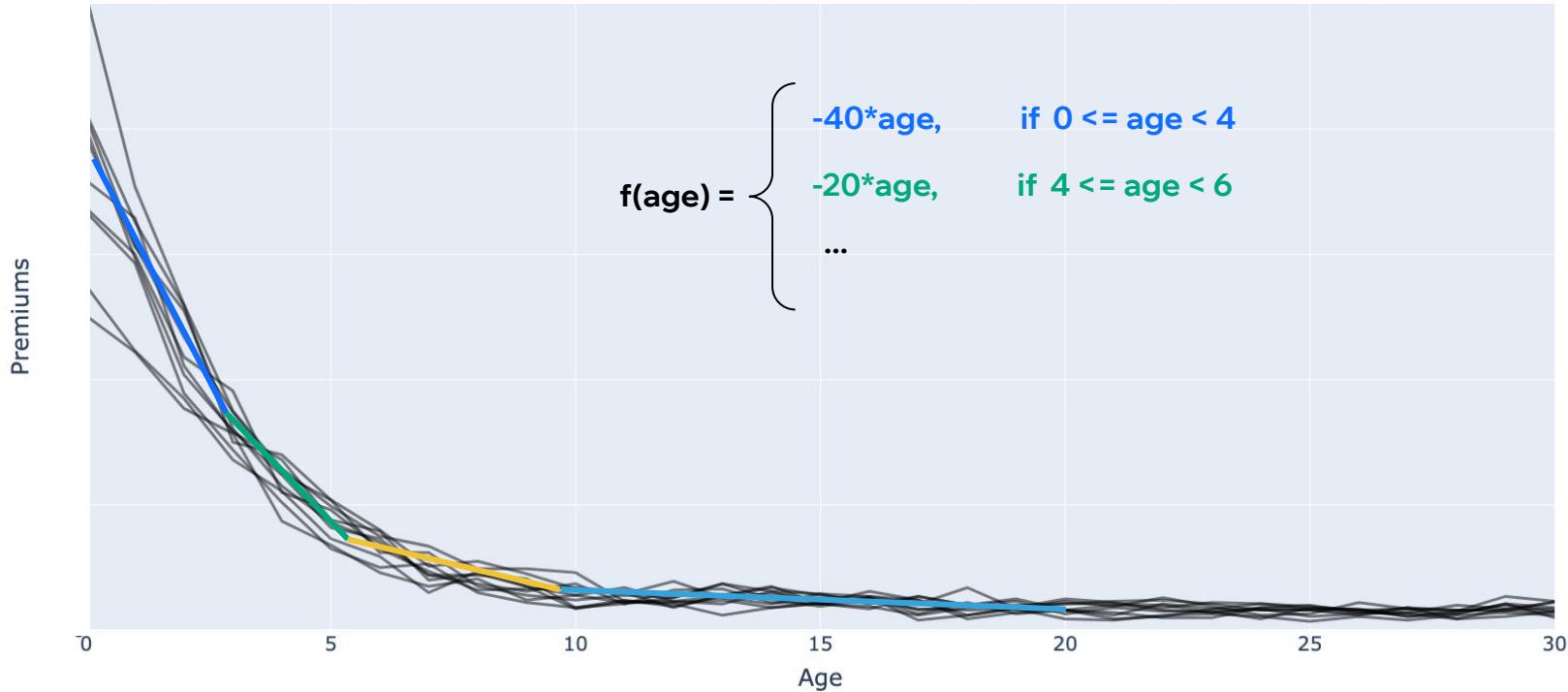
Feat engineering: decay of premiums by age

Piecewise linear function



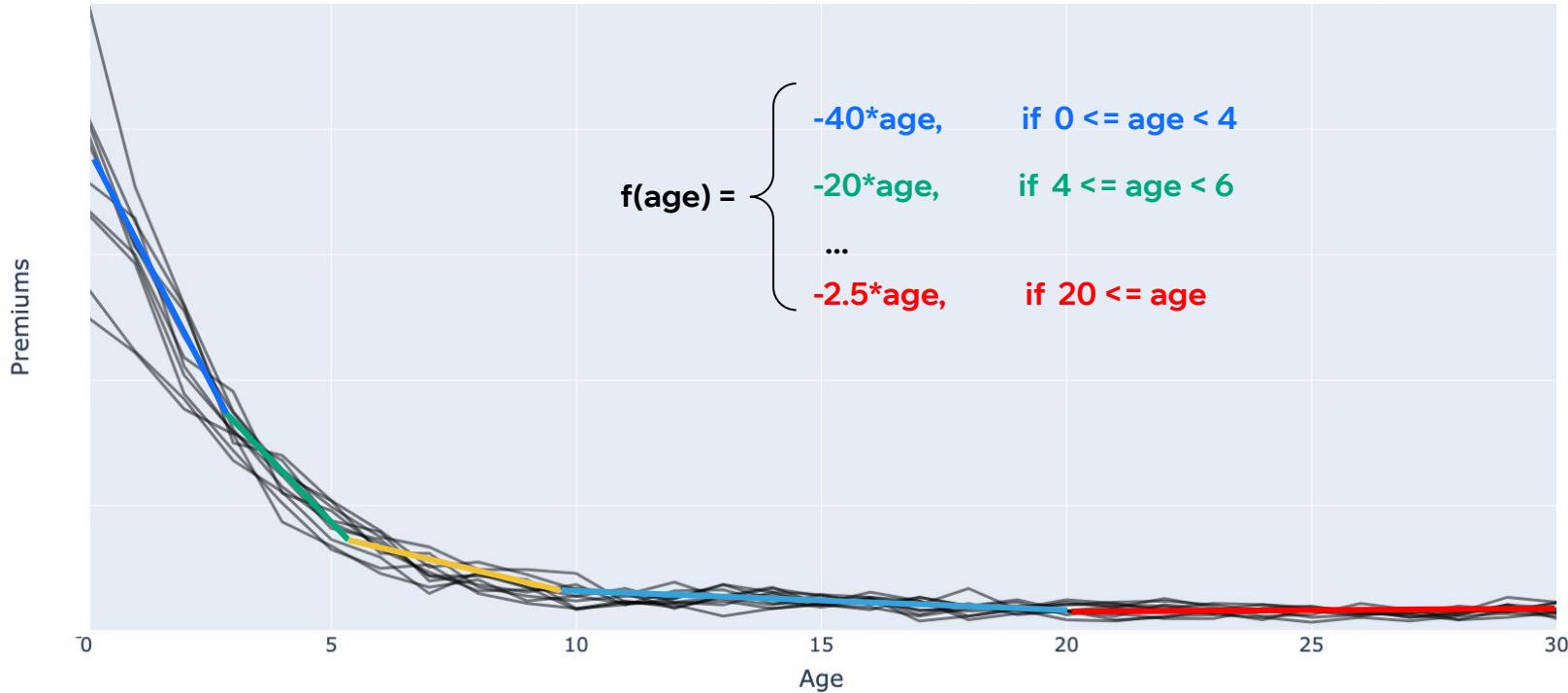
Feat engineering: decay of premiums by age

Piecewise linear function



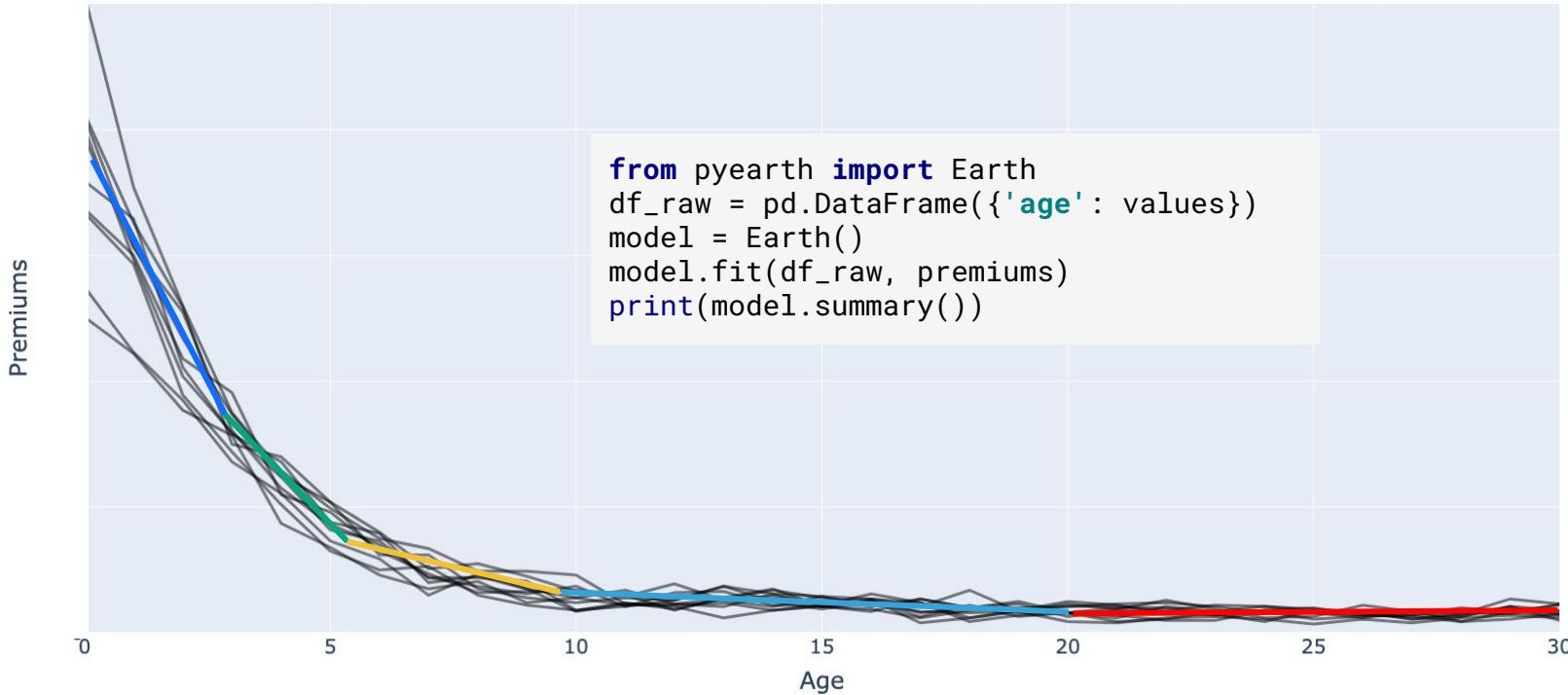
Feat engineering: decay of premiums by age

Piecewise linear function



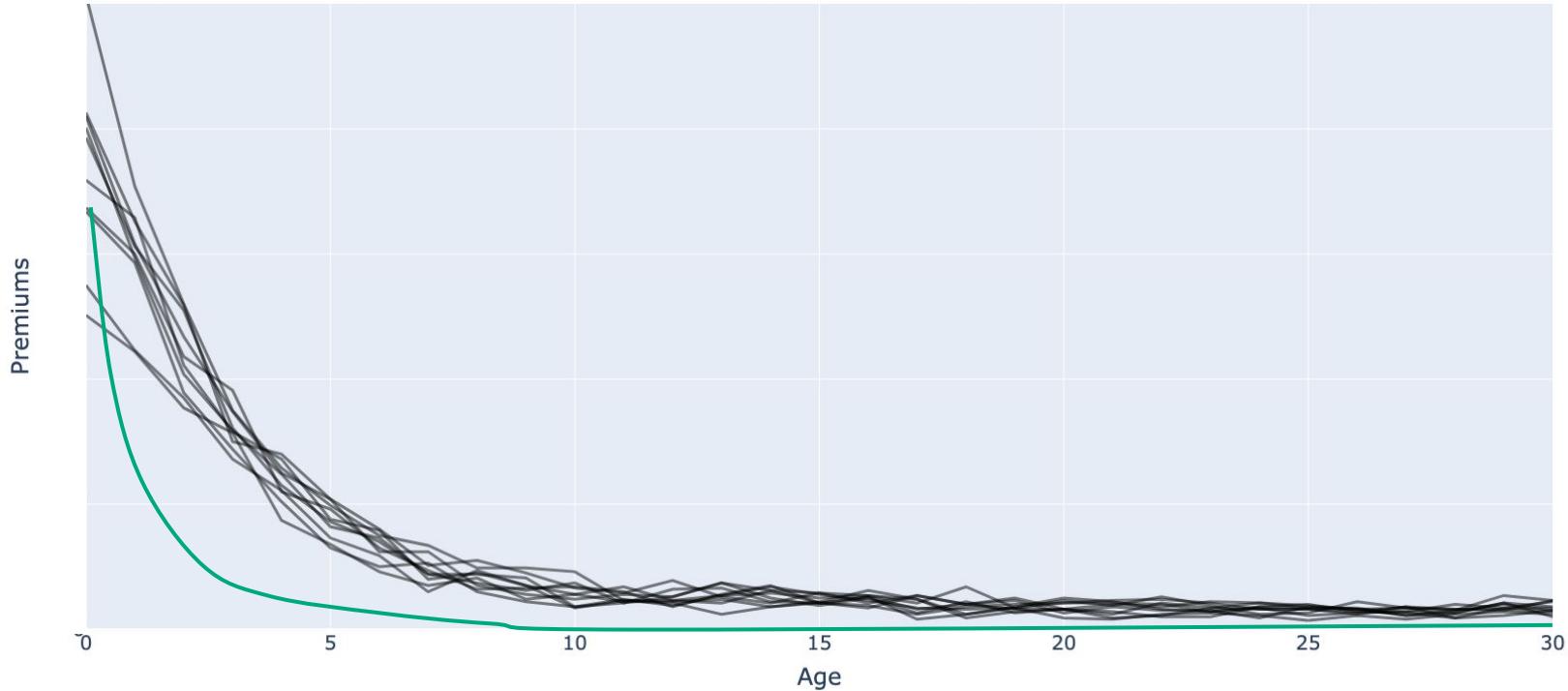
Feat engineering: decay of premiums by age

find knots with MARS from *pyearth* (python), *earth* (R)



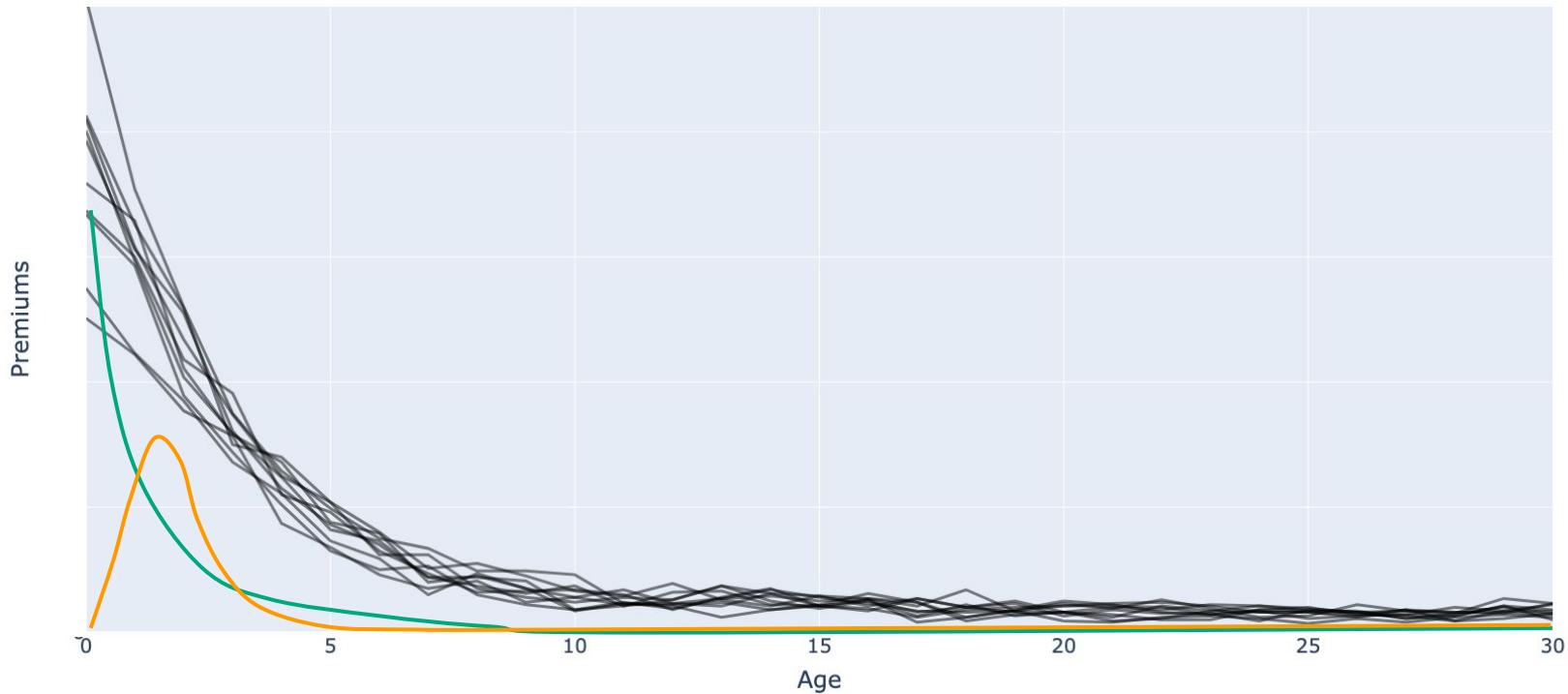
Feat engineering: decay of premiums by age

B-splines smoothing



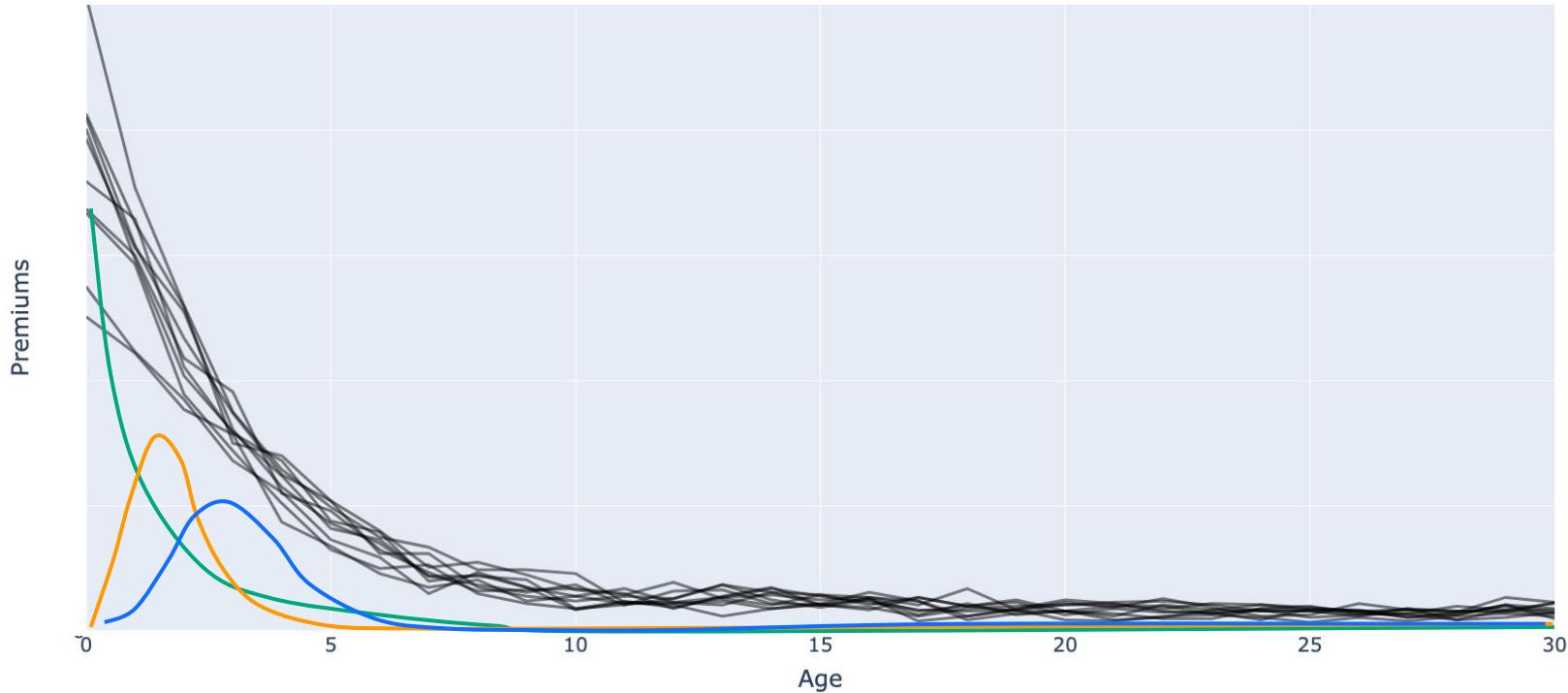
Feat engineering: decay of premiums by age

B-splines smoothing



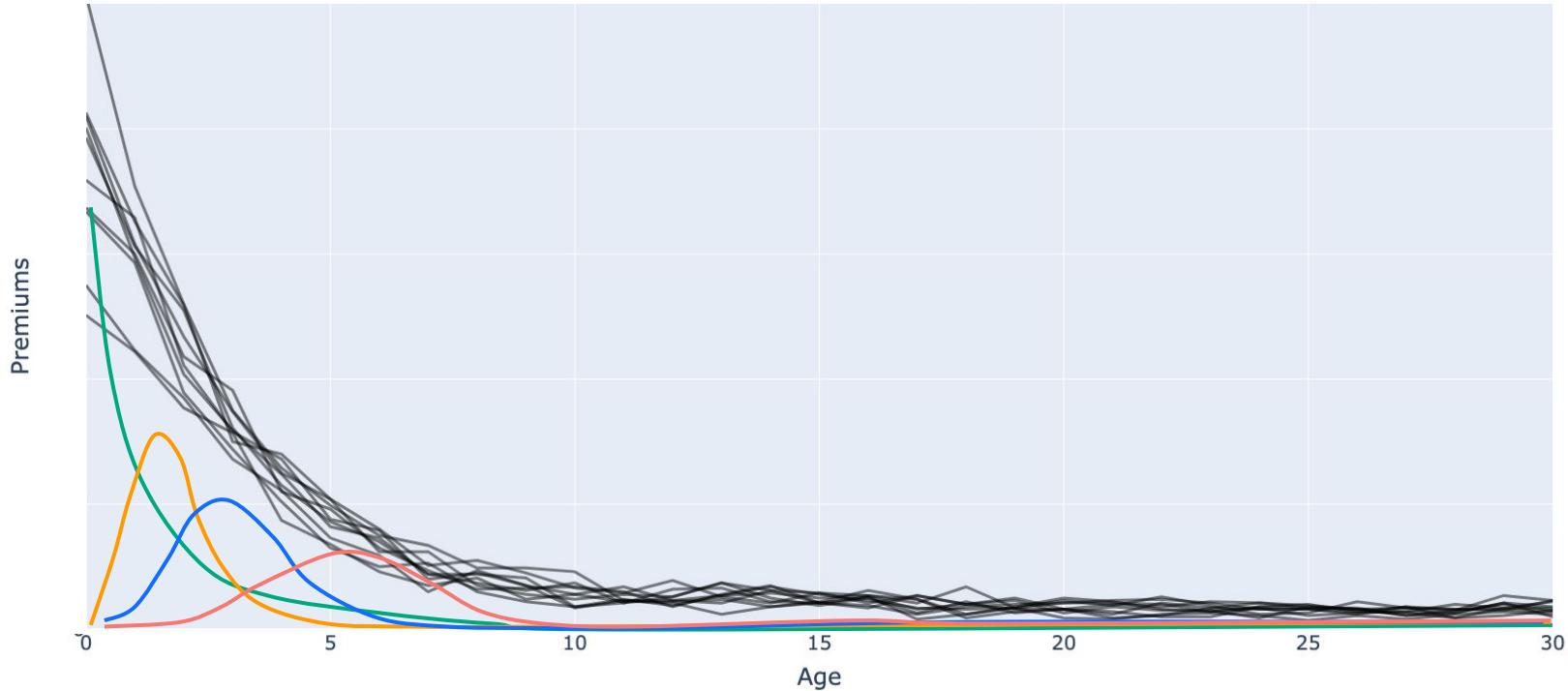
Feat engineering: decay of premiums by age

B-splines smoothing



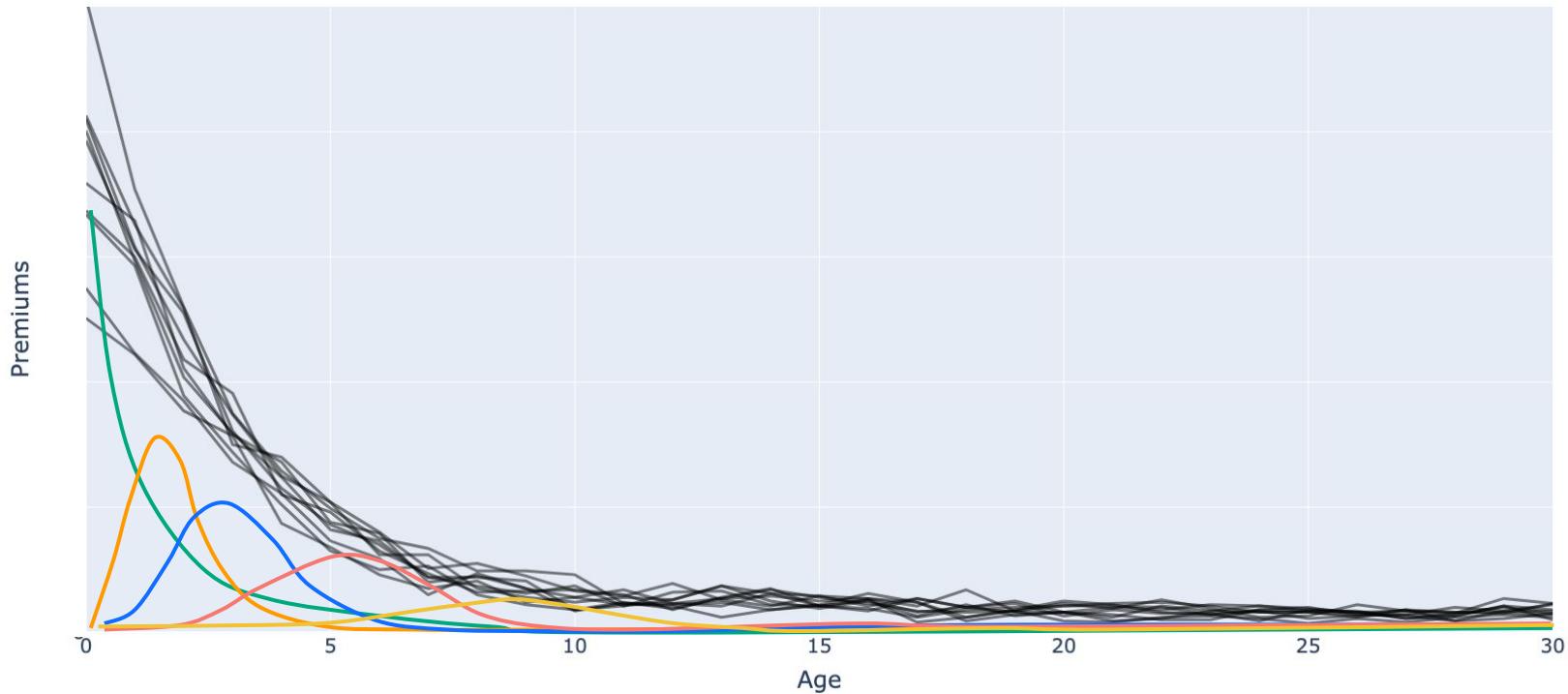
Feat engineering: decay of premiums by age

B-splines smoothing



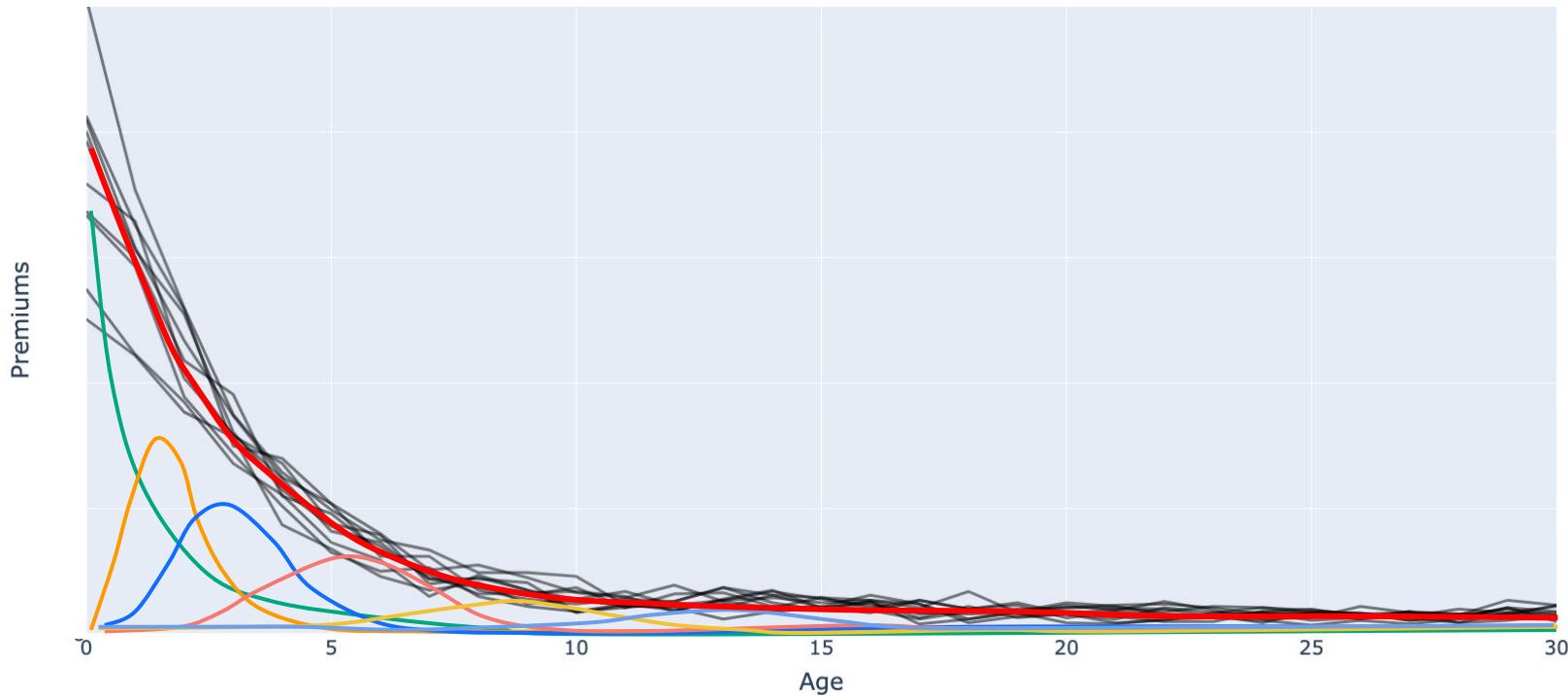
Feat engineering: decay of premiums by age

B-splines smoothing



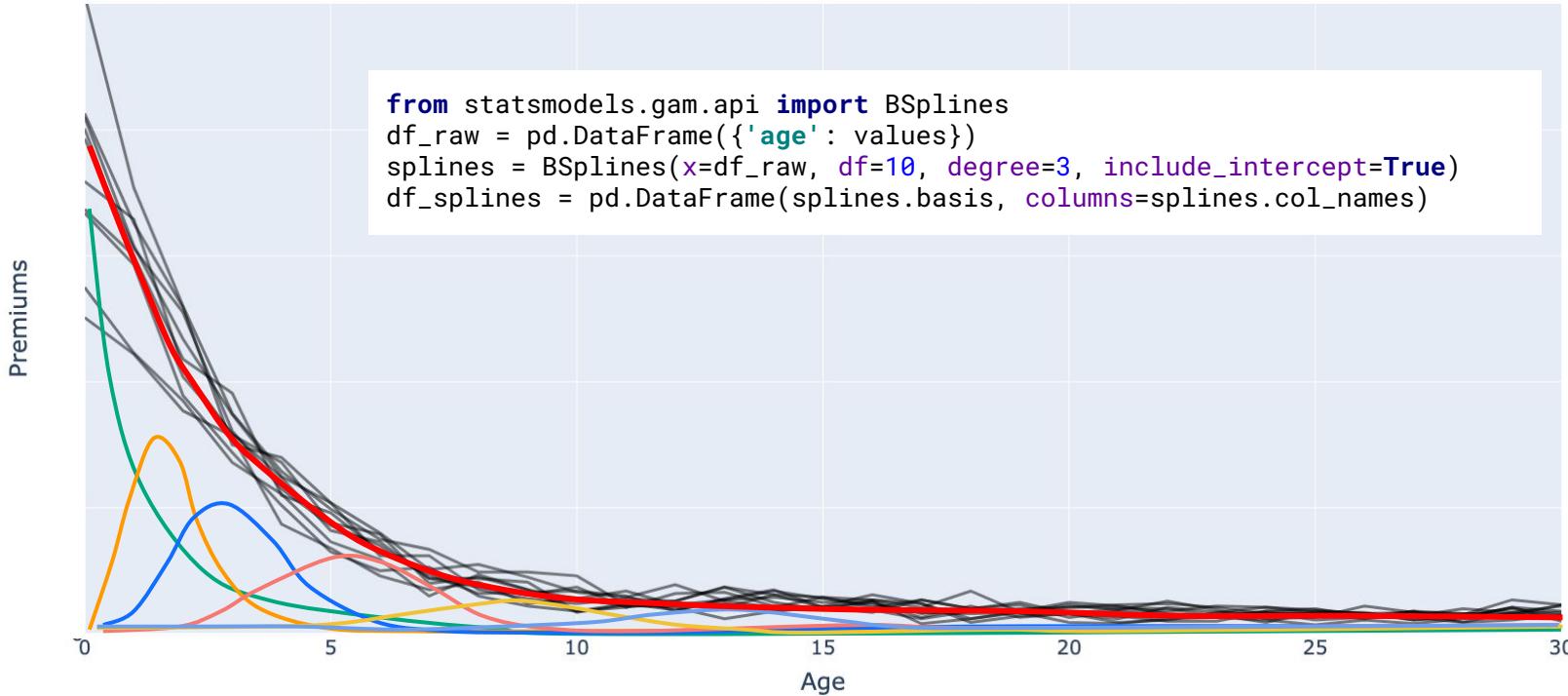
Feat engineering: decay of premiums by age

B-splines smoothing



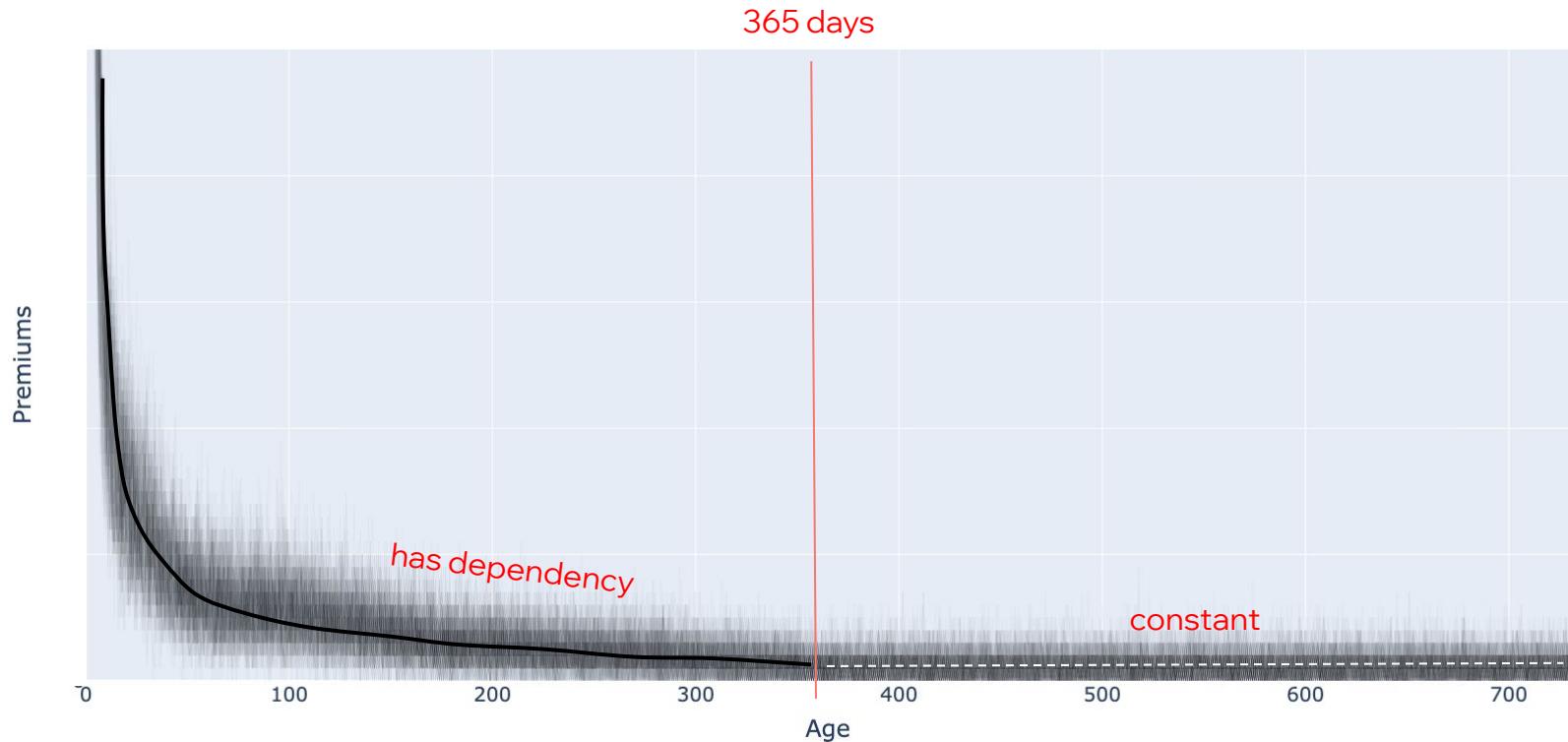
Feat engineering: decay of premiums by age

B-splines smoothing using *statsmodels* (Python)



Cut-off between Old and Recent cohorts

where premiums become constant, independent of age



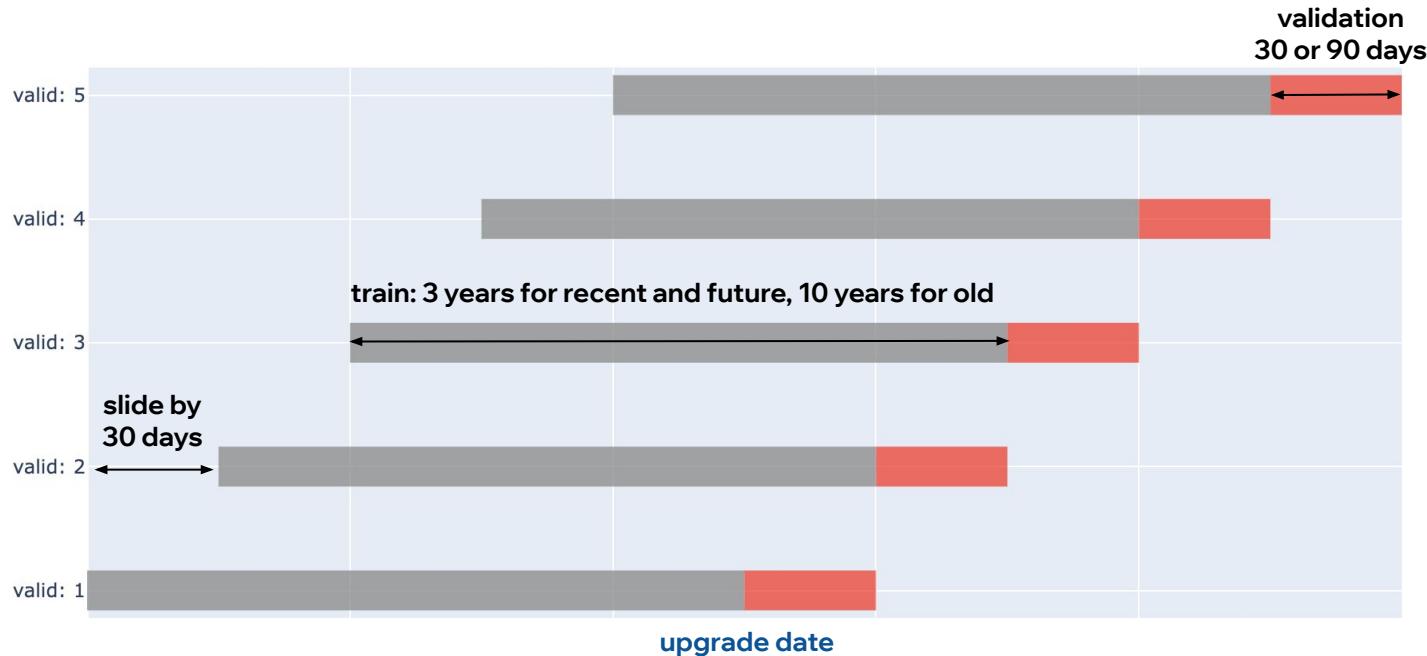
Forecast old cohorts using time series model

Good if:

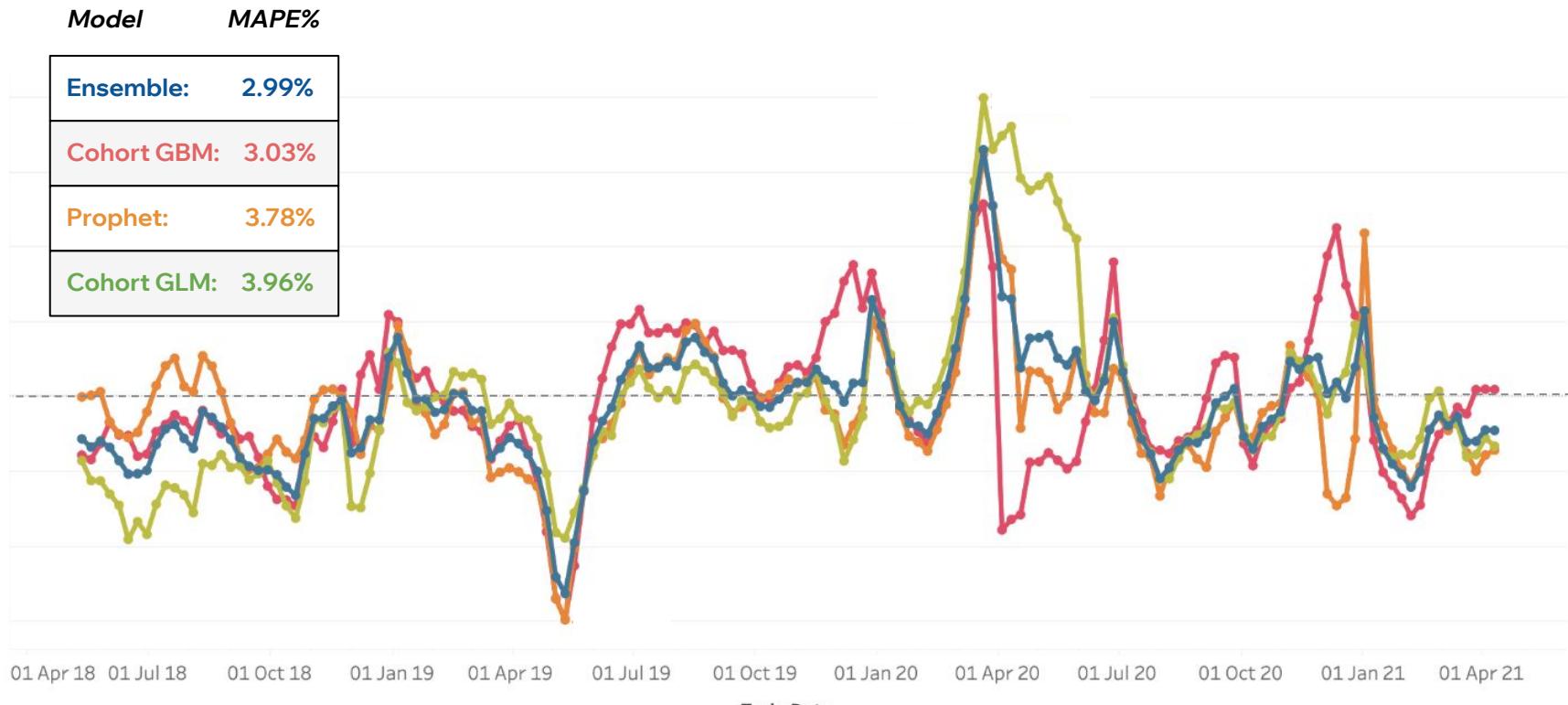
- Features joined by registration date with relevance to cohorts become insignificant for the model.
For example, the age becomes irrelevant (remember the flat tails) and the other features like cohort size (number of users registered) also become unnecessary and can be dropped without loss of forecast precision.

Validation

rolling walk forward by upgrade date



Models comparison on walk-fwd validation errors of forecasts for 30 days ahead



* this is an actual print-screen from Wix internal reports

Bonus:

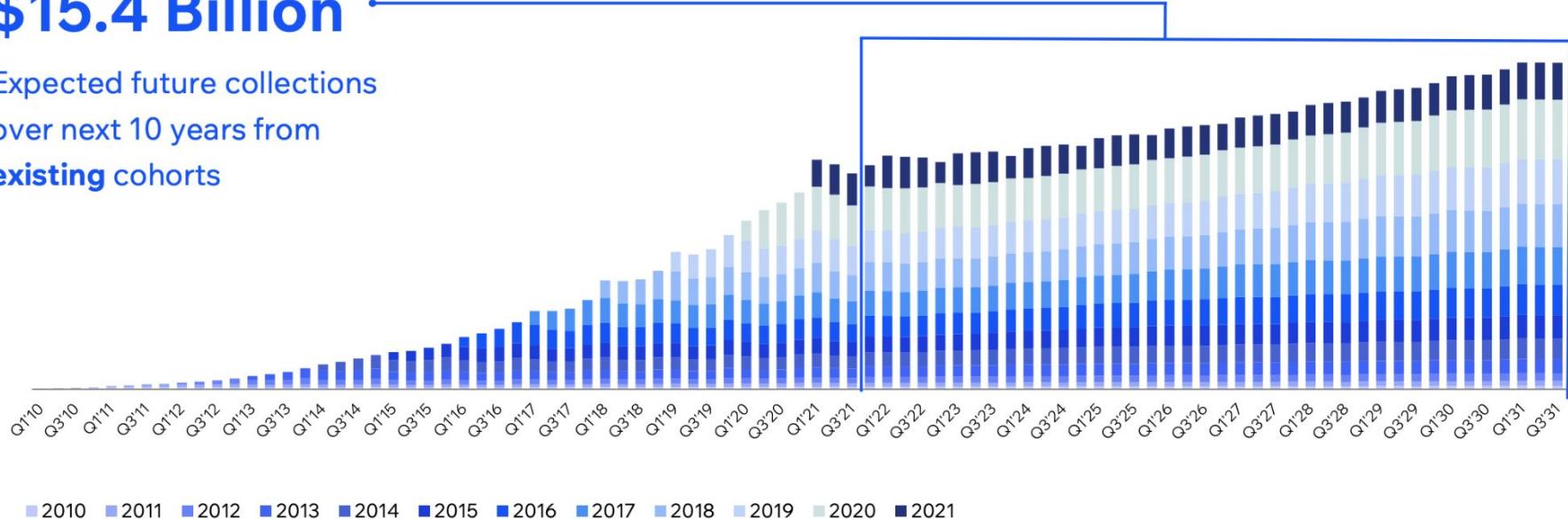
Using cohort-based models you can **aggregate** predictions **by registration date** too.
This allows understanding **which cohorts** generate **how much** out of the total forecast.

Cohort based models

can tell which cohorts will generate premiums

\$15.4 Billion

Expected future collections
over next 10 years from
existing cohorts



*as of 2021-03-31, from investor presentation, find more: <https://investors.wix.com/>

That's it.
Thank you!

Contact me:

nicolaiv@wix.com

<https://www.linkedin.com/in/nicolai-vicol/>

<https://medium.com/@nicolaivicol>

Blog about engineering & data science at Wix:

<https://medium.com/wix-engineering>

**Questions?
There are no stupid questions.
So, please ask :)**

Question:

How data was generated?

Answer:

https://github.com/nicolaivico/gen-synth-data-cohorts/blob/main/gen_data.py

```
# - multiplicative model for users:  
# users(r) = trend(r) * yearly(r) * weekly(r) * holiday(r)  
# where: r - registration date, t - upgrade date  
  
# - model for the exponential decay of premiums by age:  
# premium_rate(age) = f(age)  
# where: f(age) is a non-linear function  
  
# - Poisson model for premiums:  
# lambda(r,t) = offers(t) * trend(r) * yearly(r) * weekly(r) * premium_rate(age=t-r) * frees(r)  
# premiums(r,t) ~ Poisson(lambda(r,t))
```

Question:

How to deal with the extrapolation issue by GBMs?

Answer:

I don't have a good answer.

But maybe a simple and stupid solution is to perform the cross validation and check the bias in errors.

If any consistent bias of the same sign, adjust forecast by the percent of bias with the opposite sign.

Appendix

Time-series models

ARIMA(p, d, q)

AutoRegressive Integrated Moving Average, e.g. ARIMA(1, 1, 1)

differenced target series	autoregressive part	moving average part	error term
------------------------------	------------------------	------------------------	---------------

$$\Delta y(t) = \varphi \Delta y(t-1) + \theta \varepsilon(t-1) + \varepsilon(t)$$

p - autoregressive (AR) order

d - order of differencing

q - moving average (MA) order

Time-series models

SARIMAX(p, d, q)(P, D, Q, s)

Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors

e.g. SARIMAX(1, 1, 1)(1, 1, 1, 12)

$$\Delta y(t) = \dots + \varphi(\Delta y(t-12) + \Delta y(t-13))$$

seasonal autoregressive part

$$+ \theta(\varepsilon(t-12) + \varepsilon(t-13))$$

seasonal moving-average part

$$+ \beta X(t)$$

exogenous variables

P, D, Q - seasonal orders

s - seasonal cycle

Time-series models

Prophet, STL

$$\begin{array}{ccccc} \text{target} & \text{trend,} & \text{seasonal} & \text{holidays \&} & \text{error} \\ \text{series} & \text{"growth" term} & \text{terms} & \text{special events} & \text{term} \\ y(t) = g(t) + s(t) + h(t) + \varepsilon \end{array}$$

where: t - date (time index)

SEATS, X11

"Industry standard" used by national banks & government agencies for macroeconomic data having monthly or quarterly frequency.

LSTM

Long Short-Term Memory networks are capable of capturing patterns in the time series through sequence to sequence training and generate future sequences, i.e. forecasts.