Subsciber Analysis

SATST531 Final Project

Contents

1.Introduction **1**

2. ARIMA model **2**

2.1 data pre-process 2

2.2 model selection 3

2.3 Diagnostics 6

2.4 ARMA Conclusion 7

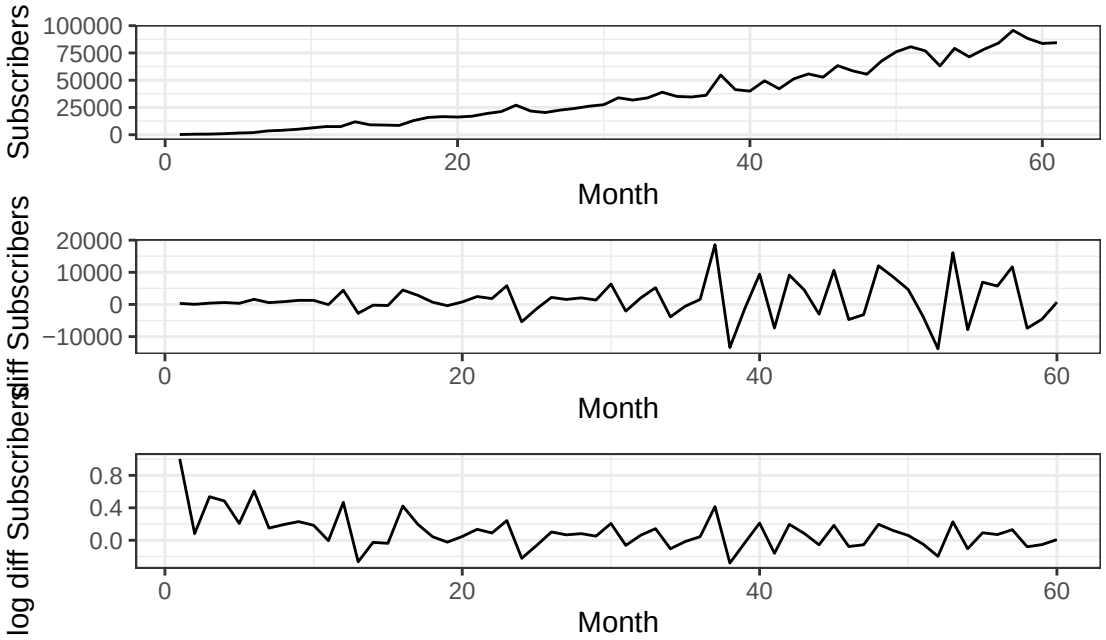
1.Introduction

Although you could spend years on the internet and never visit Twitch, however for years Twitch.tv demanded the largest portion of internet traffic. With the onset of the pandemic the streaming market burst with almost all large social media platforms such as TikTok, Facebook, Instagram, and many more to implement and offer live streaming services. Despite this, because of their advances in streaming technology and large userbase, professional streams preferred to make Twitch their home.

An increase in the demand for live streaming content has created the professional streamer. Our study focuses on the viability and long term analysis of this job. Félix Lengyel started streaming in 2014 and since then has managed to climb his way to the top of the platform. This channel was nice because of the longevity of available data for analysis found on TwitchTracker.com

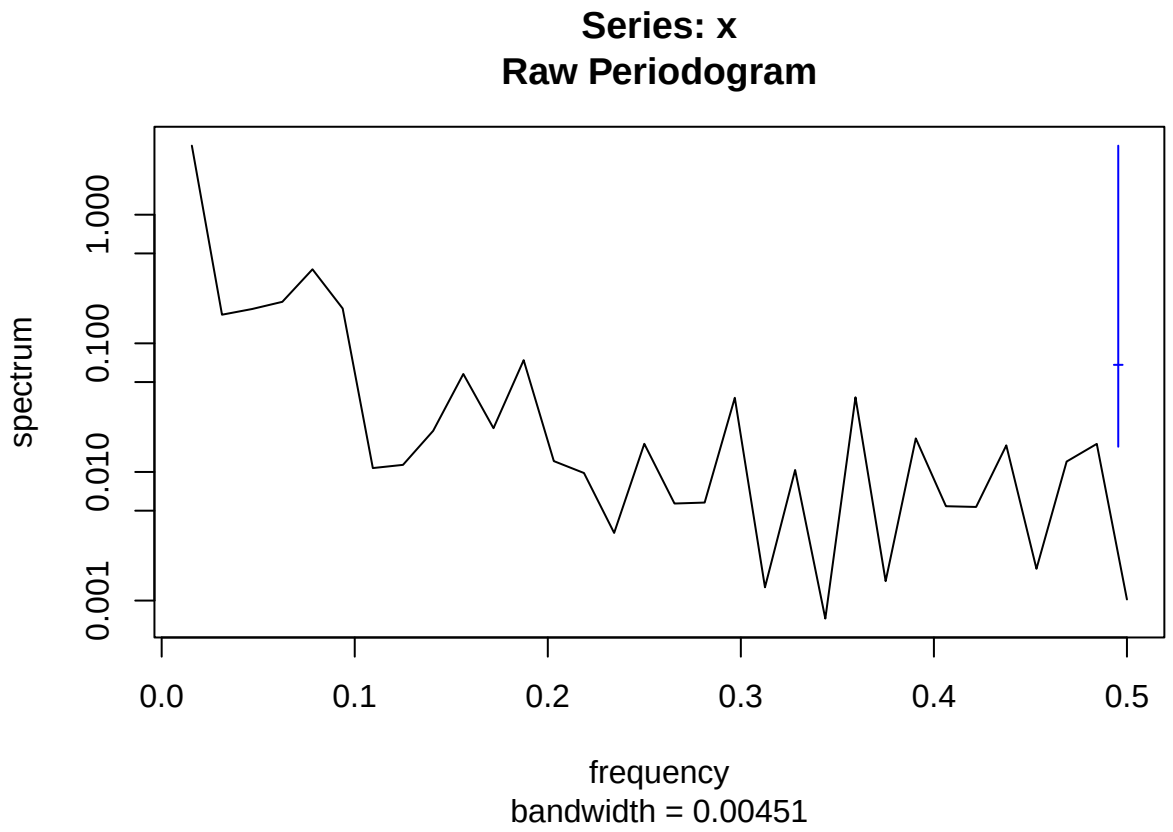
2. ARIMA model

2.1 data pre-process

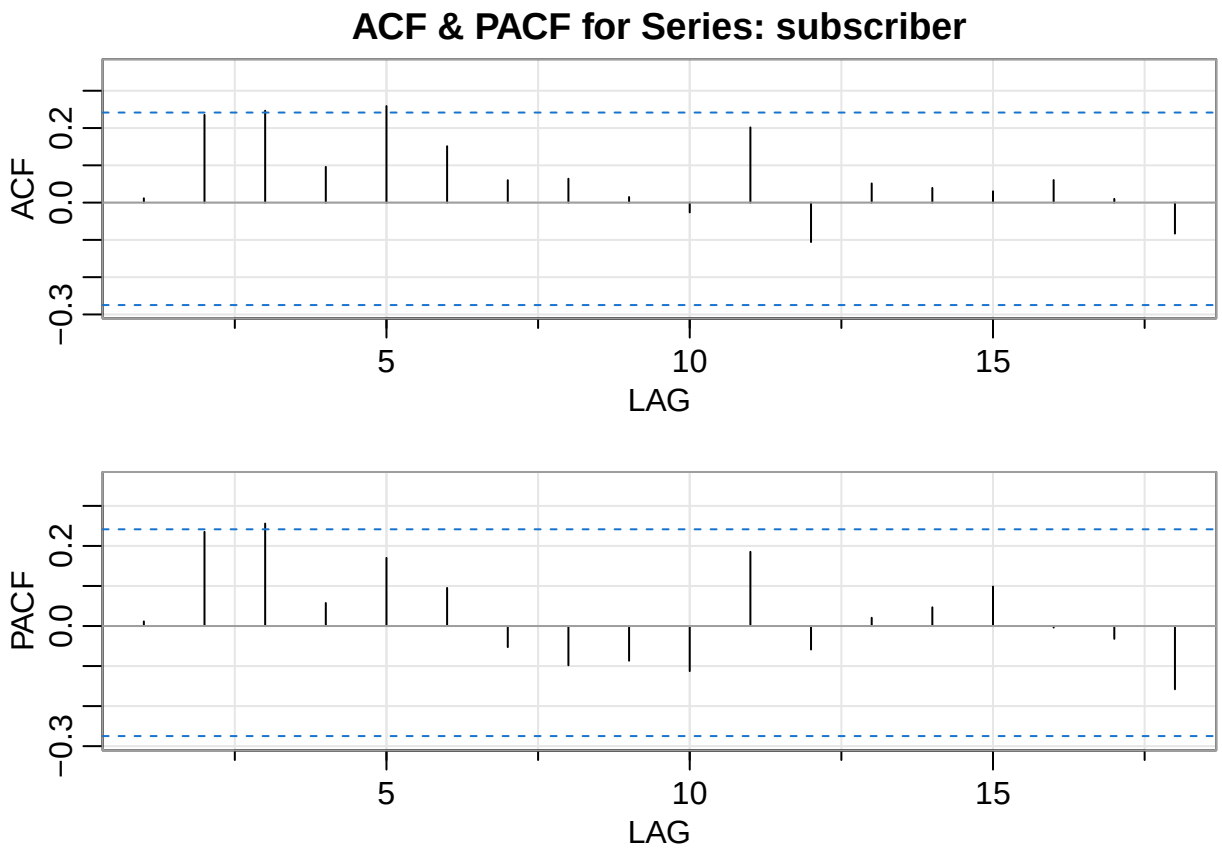


The monthly increase in subscribers is not mean stationary so that the first difference is conducted. The plot for the first difference show the sign of heteroscedasticity which can be removed by taking the logarithm. The log-diff subscriber data series look stationary and ready to apply the ARMA model.

2.2 model selection



There is no clear periodic pattern so that the seasonal attribution is not considered in the model.




```
##      [,1] [,2] [,3] [,4] [,5] [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF  0.01 0.24 0.25 0.10 0.26 0.15  0.06  0.06  0.01 -0.03  0.20 -0.11  0.05
## PACF 0.01 0.24 0.26 0.06 0.17 0.10 -0.05 -0.10 -0.09 -0.11  0.19 -0.06  0.02
##      [,14] [,15] [,16] [,17] [,18]
## ACF   0.04  0.03  0.06  0.01 -0.08
## PACF  0.05  0.10  0.00 -0.03 -0.16
```

By observing the ACF and PACF plot, it is almost white noise but some lines exceed blue dashed lines a bit which test the 95% confidence interval under the hypothesis the residuals are independent and identically distributed. Based on the ACF and PACF, p, q can be roughly estimated as $p = 0, 1, 2, 3$ and $q = 0, 1, 2, 3, 4, 5$. The possible models are conducted and compared with AIC. The grid search method is used to find the best ARIMA model:

	MA0	MA1	MA2	MA3	MA4	MA5
AR0	1.1891874	1.989055	-2.704764	-2.718568	-1.107557	-12.40532
AR1	0.9863373	-13.155179	-20.762248	-19.919574	-18.075703	-12.41512
AR2	-6.4733714	-17.448055	-20.022516	-19.140899	-17.556451	-16.65539
AR3	-13.3322207	-16.896406	-18.023058	-17.715896	-15.230672	-16.90922

Among all possible parameters attempted, the model ARIMA$(1, 1, 2)$ is the best with the smallest AIC.

The coefficients of the fitted model with `AR=1` and `MA=2` are:

Dependent variable

Predictors

Estimates

CI

p

ar1

0.99

0.95 – 1.02

<0.001

ma1

-1.26

-1.54 – -0.99

<0.001

ma2

0.52

0.23 – 0.80

<0.001

Observations

60

R2

0.983

By checking the corresponding roots for `AR` and `MA` respectively:

Table 2: AR

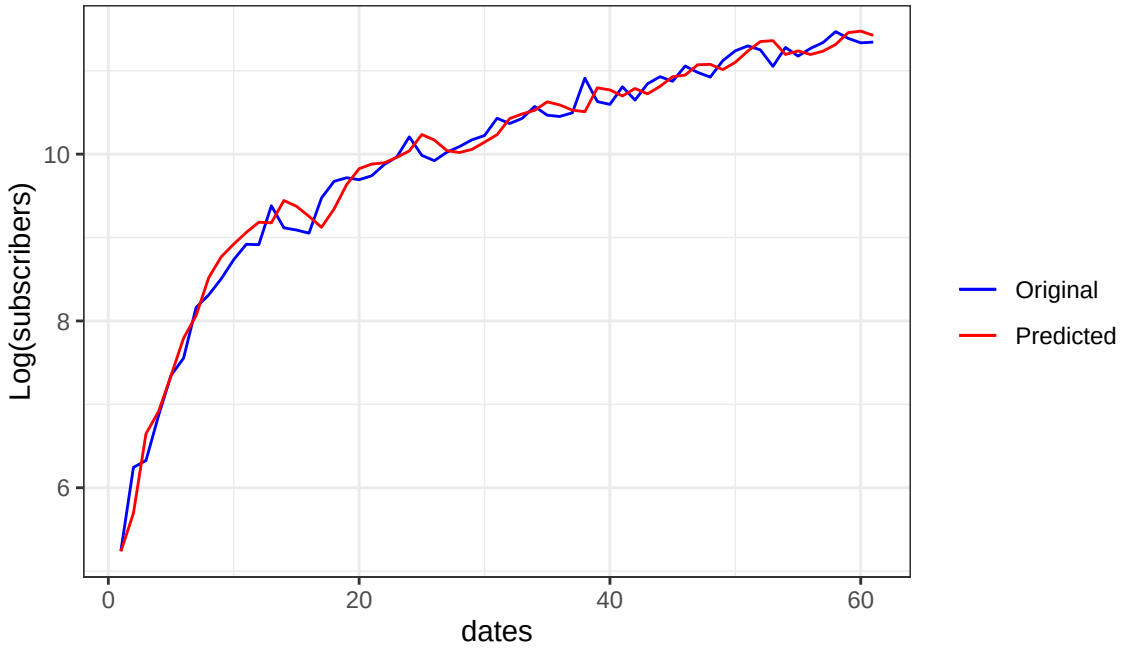
x
1.01363+0i

Table 3: MA

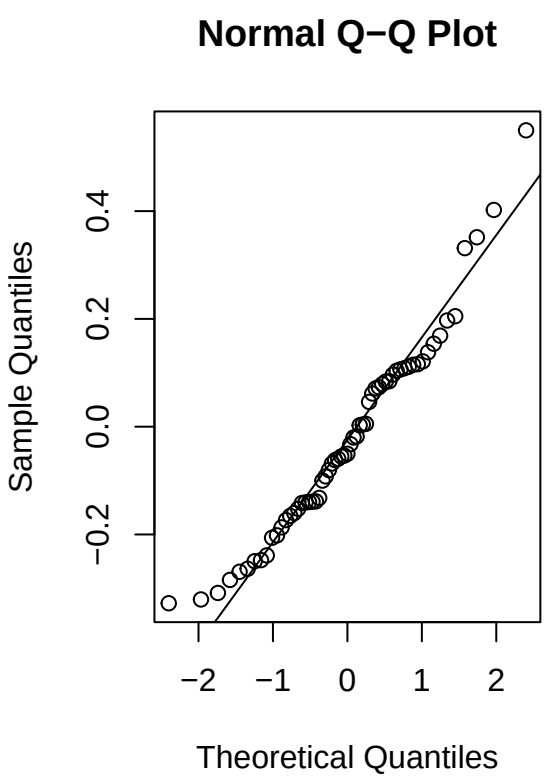
x
1.219695+0.666131i
1.219695-0.666131i

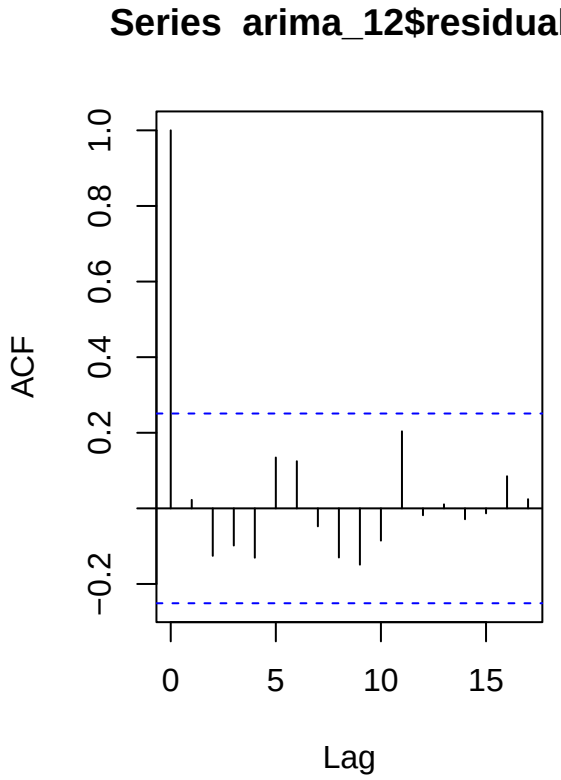
It is found that all roots are outside the unit circle but the root for AR is actually very close to the unit circle which indicates that the monthly increase in subscribers is an almost stationary growth process.

2.3 Diagnostics



From the plot, the original and predicted data highly agree with each other. By checking the residuals, it's observed that they are well normally distributed and ACF plot shows a white noise process as assumed [1].





2.4 ARMA Conclusion

Although the model revealed no seasonality, the best model of $ AR 1 MA 2$ found an increasing trend, promising relevant hopeful results from pomp analysis

Now, we will try fit the data with a pomp model.

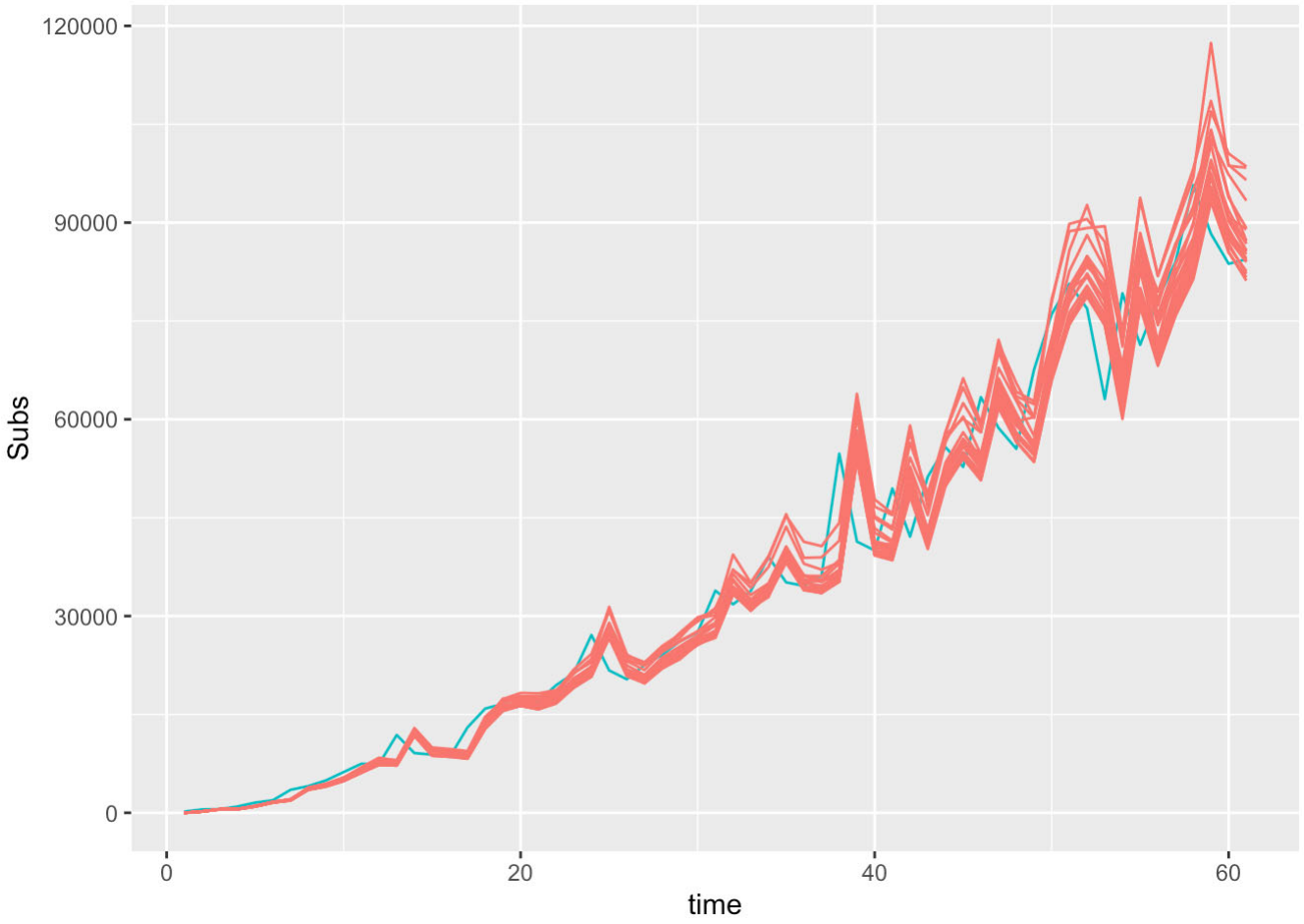
The following is the code for the model:

```
library(pomp)
library(tidyverse)
data = read.csv('twitch.csv')

covar <- covariate_table(
  time=data$time,
  S=lag(data$Subscribers, 1, default=0),
  times = "time"
)

read_csv(paste0('twitch.csv')) %>%
  select(time=time, Subs=Subscribers, View=AvgVeiwers) -> measBVS

# B: beginning population
# V: Viewers (a temporary container)
# S: Subscribers
# N: total number of users
#
# Beta: Beginning -> Viewers
# mu_VS: Viewers -> Subscribers
# mu_SB: Subscribers -> Beginning
# Beta_sigma determines random walk of Beta
#
#
bvs_step <- Csnippet("
  Beta=expit(logit(Beta)+rnorm(0, Beta_sigma));
  D=rbinom(S,1-exp(-mu_SB));
  ")

bvs_rinit <- Csnippet("
  Beta=Beta_0;
  D=0;
  ")

bvs_dmeas <- Csnippet("
  double Views = rbinom(N-S,1-exp(-Beta*S/N));
  lik= dnorm(Subs+D-S, Views*(1-exp(-mu_VS)), Views*(1-exp(-mu_VS))/10, 1);
  if (lik>0) {
    lik=0;
  }
  if (lik<-100) {
    lik=-100;
  }
  ")

bvs_rmeas <- Csnippet("
  double Views = rbinom(N-S,1-exp(-Beta*S/N));
  View = Views/30;
  Subs= rnorm(Views*(1-exp(-mu_VS)), Views*(1-exp(-mu_VS))/10) + S - D;
  ")
```

```
partrans <- parameter_trans(
  log=c("Beta_sigma"),
  logit=c("mu_VS","mu_SB", "Beta_0")
)

measBVS %>%
  pomp(rprocess=euler(bvs_step,delta.t=1),
       times="time",
       t0=1,
       statenames=c("Beta","D"),
       paramnames=c("Beta_sigma","mu_VS", "mu_SB", "N", "Beta_0"),
       partrans=partrans,
       covar=covar,
       rinit=bvs_rinit,
       rmeasure=bvs_rmeas,
       dmeasure=bvs_dmeas
       ) -> measBVS
```

In summary, this is the idea behind the model. There are three compartments: Beginning, Viewers, and Subscribers. Every month, there is a certain proportion of “Beginners” who become “Viewers”, and similarly for Viewers to Subscribers, and Subscribers back to beginners (people who unsubscribe, for example). In addition, Viewers is not a cummulative category, but gets reset to 0 every month.

The following is what a simulation of the data looks like for some parameter values.



Now, lets use iterated filtering for this model:

```
library(doParallel)
registerDoParallel(makePSOCKcluster(detectCores()))
foreach(i=1:20,.combine=c) %dopar% {
  library(tidyverse)
  library(pomp)
  measBVS %>%
      mif2(
        params=c(Beta_sigma=0.2,mu_VS=0.37,mu_SB=0.05,Beta_0=0.2,N=41500000),
        Np=2000, Nmif=100,
        cooling.fraction.50=0.5,
        rw.sd=rw.sd(Beta_sigma=0.2,mu_VS=0.37,mu_SB=0.05,Beta_0=ivp(0.2)),
        paramnames=c("Beta_sigma","mu_VS", "mu_SB", "Beta_0","N")
      )
} -> mifs_local
```

```
set.seed(123456)
runif_design(
  lower=c(Beta_sigma=0.01,mu_VS=0.01,mu_SB=0.01,Beta_0=0.01),
  upper=c(Beta_sigma=0.5,mu_VS=0.5,mu_SB=0.5,Beta_0=0.5),
  nseq=40
) -> guesses
mf1 <- mifs_local[[1]]
```

```
library(doParallel)
registerDoParallel(makePSOCKcluster(detectCores()))
foreach(guess=iter(guesses,"row"), .combine=rbind) %dopar% {
  library(pomp)
  library(tidyverse)
  try({
  mf1 %>%
      mif2(Nmif=25, params=c(guess,fixed_params)) %>%
      mif2(Nmif=50) -> mf
    replicate(
  8,
      mf %>% pfilter(Np=4000) %>% logLik()
    ) %>%
      logmeanexp(se=TRUE) -> ll
    mf %>% coef() %>% bind_rows() %>%
  bind_cols(loglik=ll[1],loglik.se=ll[2])
  }, silent=TRUE)
} -> results
```

The likelihood for this model is the following:

```
## -866.0623
```

Given the results of the simulation and the log likelihood, it seems that the ARMA model performs better than

the pomp model, and that the pomp model relies heavily on the subscriber count from the previous month to predict that of the current month. It is possible, though, that a different pomp model may have worked out better.