R Notebook

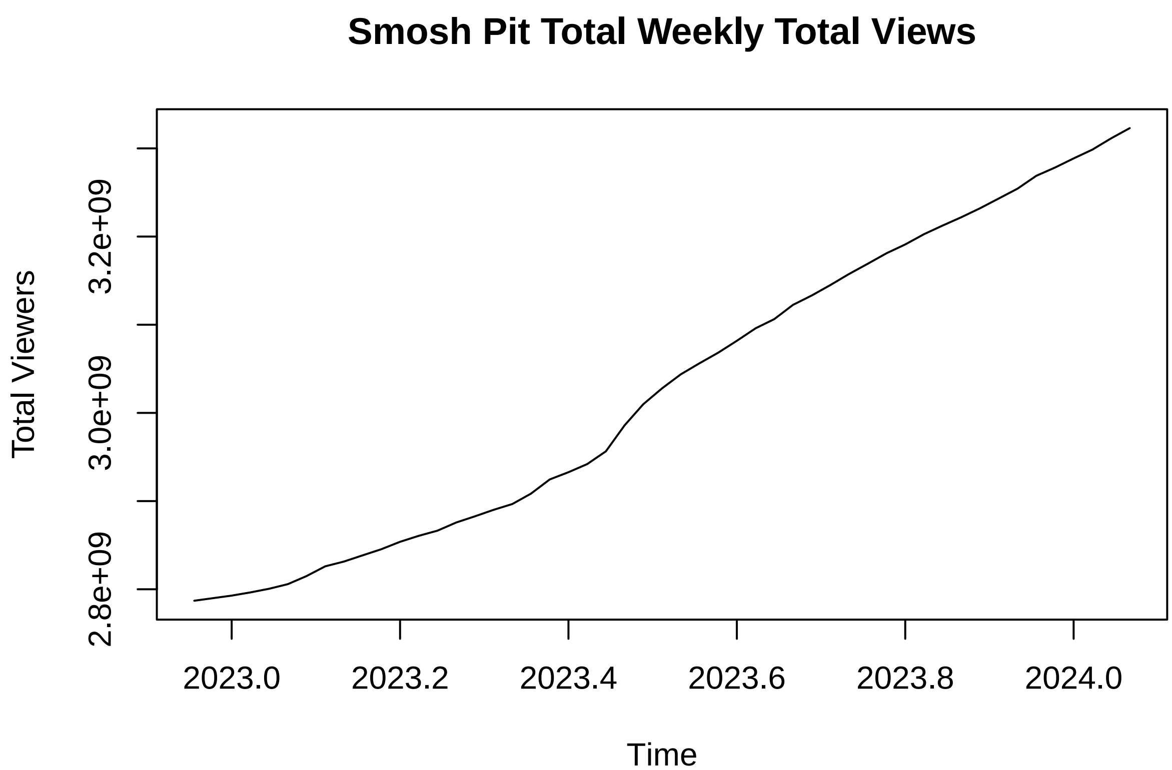
```r
#install.packages("dplyr")
#install.packages("tidyverse")
#install.packages("astsa")
#install.packages("forecast")

library(dplyr)
library(tidyverse)
library(astsa)
library(forecast)
```

```r
PATH <- ""
smosh <- read.csv(paste(PATH, "smosh_market.csv", sep = ""))
head(smosh)
```

```
##   Day    Month Year Total.Viewers
## 1  25 December 2022    2787011608
## 2   2  January 2023    2789909808
## 3  10  January 2023    2792792592
## 4  18  January 2023    2796442297
## 5  26  January 2023    2800576231
## 6   3 February 2023    2805812495
##                                                                                 Video.1
## 1 Try Not To Laugh Challenge #110 - Gauntlet w/ Our Crew! (Dec. 20 2022; 1699048, 24:05)
## 2                            Eat It Or Yeet It: Cast vs. Crew! (Dec. 27 2022; 896884, 25:49)
## 3                              Try Not To Laugh 2022 Marathon (Jan. 3 2023; 804787, 8:23:46)
## 4                         How Much Do We Know About Spongebob? (Jan. 12 2023; 692537, 21:11)
## 5                                  Short Kings Rank Short Kings (Jan. 19 2023; 684072, 36:59)
## 6                               Try Not To Laugh Challenge #112 (Jan. 31 2023; 1457075, 19:58)
##                                                                                          Video.2
## 1                         These Memes Destroyed Us (Who Meme'd It) (Dec. 22, 2022; 1422918, 28:13)
## 2                                        Beopardy 2022 Marathon (Dec. 29 2022; 789801, 04:57:38)
## 3                       Reading Unhinged Reddit Stories w/ MacDoesIt (Jan. 5 2023; 2515511, 45:47)
## 4                                     Try Not To Laugh Challenge #111 (Jan. 17 2023; 2693844, 26:05)
## 5                      Filipino Food Taste Test (Eat It or Yeet It) (Jan. 24 2023; 1970618, 25:04)
## 6 We Try The TikTok Candle Challenge... and MORE! | The Challenge Pit (Feb. 2 2023; 818879, 22:07)
##                                                          Video.3 Video.4
## 1                                                          ; 0,    ; 0,
## 2                                                          ; 0,    ; 0,
## 3 Eat It Or Yeet It 2022 Marathon (Jan. 10 2023; 548572, 7:10:39)    ; 0,
## 4                                                          ; 0,    ; 0,
## 5             Can I Work A Real Job? (Jan. 26 2023; 864940, 30:33)    ; 0,
## 6                                                          ; 0,    ; 0,
##   Total.Views.of.Videos.Posted.that.Week X..Videos.Posted
## 1                                3121966                2
## 2                                1686685                2
## 3                                3868870                3
## 4                                3386381                2
```

```
## 5                                3519630                3
## 6                                2275954                2
##   Avg..Views.of.Videos.Posted.That.Week
## 1                               1560983
## 2                                843343
## 3                               1289623
## 4                               1693191
## 5                               1173210
## 6                               1137977
##   Total.Duration.of.Videos.Posted.That.Week
## 1                                        NA
## 2                                        NA
## 3                                        NA
## 4                                        NA
## 5                                        NA
## 6                                        NA
```

```
smosh <- smosh %>% rename("Total Viewers" = "Total.Viewers")
sm <- ts(smosh["Total Viewers"], start = c(2022, 44), frequency = 45)
plot(sm, main = "Smosh Pit Total Weekly Total Views")
```



- Non-stationary in mean
- Dataframe only about a year, likely won't track seasonality (if any) and unlikely due to nature of channel
- Mild viewership jump around June 2023 (steeper slope after jump?)

```
growth <- diff(sm)

which.max(growth)
```

```
## [1] 23
```

```r
growth[23]
```

```
## [1] 29457404
```

```r
which.min(growth)
```

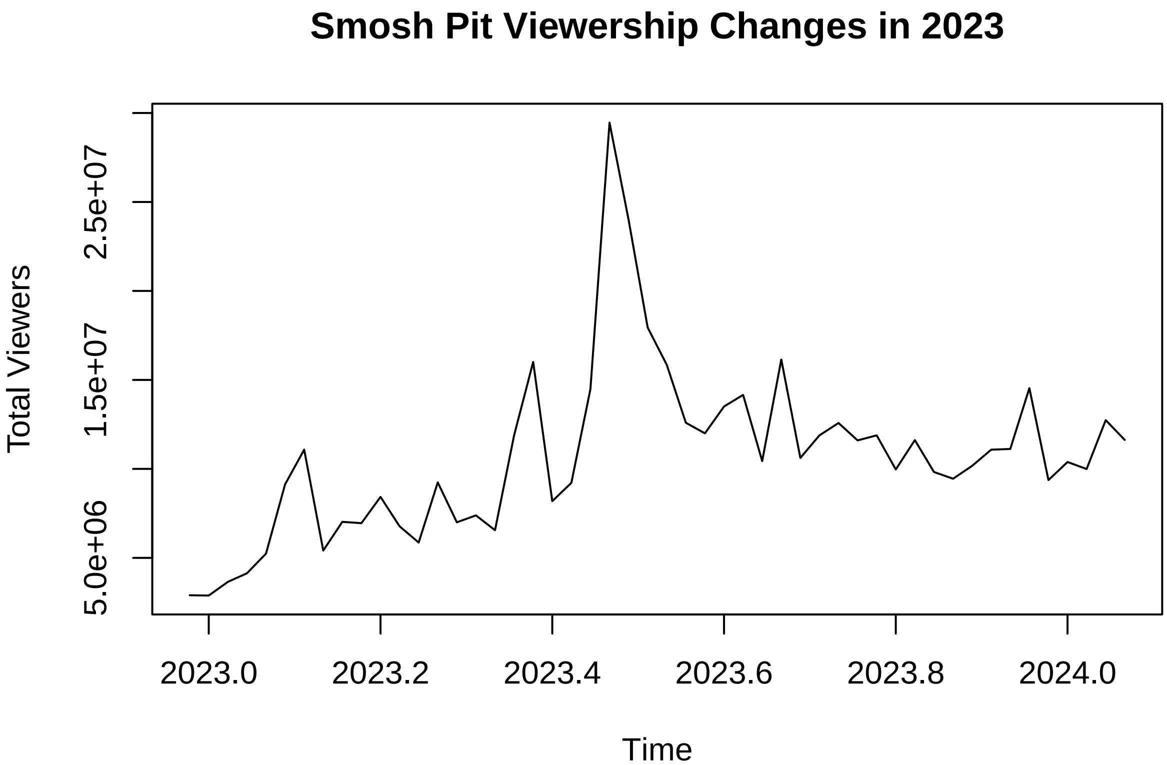
```
## [1] 2
```

```r
growth[2]
```

```
## [1] 2882784
```

```r
# 24 and 25 (now 23 and 24) have most significant increases in viewership
# (Anthony returns - 24th Anthony specific videos (in title))
```

```r
plot(growth, type = "l", main = "Smosh Pit Viewership Changes in 2023")
```




```r
# 5 noticeable peaks (7, 18/19, 23/24, 32, 45)
# post big peak - different mean, variance pretty constant aside from peaks
```

- Mean more stationary…how significant is difference after massive peak
- Variance changes: some kind of seasonality or caused by outside events (like major peak)?? - appear about eqidistant of eachother

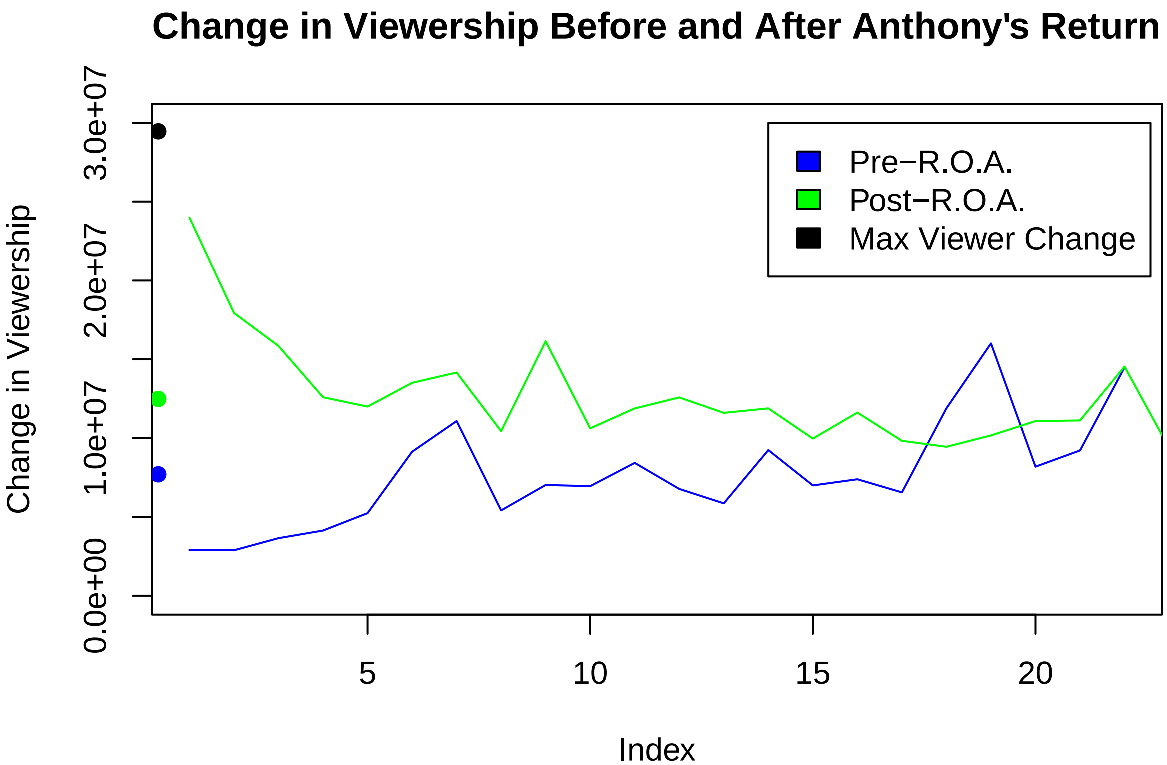
```r
# roa = return of Anthony

pre_roa <- growth[1:22]
post_roa <- growth[24:50]
mean(pre_roa)
```

```
## [1] 7701073
```

```r
mean(post_roa)
```

```
## [1] 12483718
```

```r
plot(pre_roa, type = "l", col = "blue", ylim = c(0, 30000000), ylab = "Change in Viewership", main = "Ch
lines(post_roa, type = "l", col = "green")
points(x = c(0.3, 0.3, 0.3), c(mean(pre_roa), mean(post_roa), max(growth)), col = c("blue", "green", "bl
legend(14, 30000000, legend = c("Pre-R.O.A.", "Post-R.O.A.", "Max Viewer Change"), fill = c("blue", "gre
```



Post return of Anthony has a sustained increase in viewership. Jump at end of pre-ROA period and dip at end of post-ROA period - is time series stabilizing around specific mean or residual effect of week 19 peak?

No apparent trend before or after Anthony's return. Potential seasonality? Effect of intervention mimiking autocorrelation?

```r
video1_len <- strcapture(".*, ([0-9]*:)*([0-9]*):([0-9]*)).*", smosh$Video.1,
                         list(hour_1 = "", minute_1 = "", second_1 = ""))
video2_len <- strcapture(".*, ([0-9]*:)*([0-9]*):([0-9]*)).*", smosh$Video.2,
                         list(hour_2 = "", minute_2 = "", second_2 = ""))
video3_len <- strcapture(".*, ([0-9]*:)*([0-9]*):([0-9]*)).*", smosh$Video.3,
                         list(hour_3 = "", minute_3 = "", second_3 = ""))
video4_len <- strcapture(".*, ([0-9]*:)*([0-9]*):([0-9]*)).*", smosh$Video.4,
                         list(hour_4 = "", minute_4 = "", second_4 = ""))

video_lengths <- cbind(video1_len, video2_len, video3_len, video4_len)
video_lengths
```

Finish formatting dataframe

```
##    hour_1 minute_1 second_1 hour_2 minute_2 second_2 hour_3 minute_3 second_3
```

```
## 1               24       05              28       13   <NA>     <NA>     <NA>
## 2               25       49    04:       57       38   <NA>     <NA>     <NA>
## 3      8:       23       46              45       47     7:       10       39
## 4               21       11              26       05   <NA>     <NA>     <NA>
## 5               36       59              25       04                30       33
## 6               19       58              22       07   <NA>     <NA>     <NA>
## 7               21       26              50       40   <NA>     <NA>     <NA>
## 8               18       12              30       07   <NA>     <NA>     <NA>
## 9               20       10              24       35               0       54
## 10              26       46              24       25     1:       13       45
## 11              21       46              21       15                20       53
## 12              30       20     1:       14       41                18       13
## 13              17       11              20       08                17       24
## 14     1:       16       19              24       52                27       44
## 15              17       10              20       34     1:       05       10
## 16              24       41              39       04                29       37
## 17              17       05              20       54     1:       14       29
## 18              42       30     1:       03       53                20       17
## 19              30       33     1:       16       59                19       05
## 20              38       44              14       44                28       45
## 21     1:       15       45              18       08                18       06
## 22              24       18              21       55     1:       09       57
## 23              18       28              40       10     1:       14       30
## 24              26       03              39       30     1:       10       16
## 25              21       31     1:       14       26                19       43
## 26              24       08     1:       08       15                31       49
## 27     1:       10       33              18       55                24       23
## 28     1:       13       10              25       47                26       41
## 29              25       36              36       22     1:       08       02
## 30              24       47              23       07     1:       10       43
## 31              22       10              45       19     1:       17       35
## 32              28       26     1:       14       05                26       33
## 33              49       07     1:       09       39                27       36
## 34     1:       15       59              22       32                28       34
## 35              26       55              30       31     1:       10       32
## 36              23       28              42       37     1:       16       39
## 37              22       22              26       34     1:       22       38
## 38              17       27              37       43     1:       08       02
## 39              22       28     1:       15       04                22       20
## 40              48       09     1:       22       43                26       01
## 41     1:       05       11              19       19                36       03
## 42     1:       07       41              30       42                48       29
## 43              25       01              48       12                55       32
## 44              27       09     1:       15       57   <NA>     <NA>     <NA>
## 45              19       31              34       56     1:       10       02
## 46              35       02     1:       08       20                27       19
## 47     1:       15       24     1:       20       00                31       38
## 48     1:       18       32     8:       34       13     4:       44       32
## 49     6:       28       27     9:       06       33                33       57
## 50              14       50              29       31     1:       21       26
## 51              51       29              33       31     1:       13       23
##    hour_4 minute_4 second_4
## 1    <NA>     <NA>     <NA>
## 2    <NA>     <NA>     <NA>
```

```
## 3    <NA>     <NA>     <NA>
## 4    <NA>     <NA>     <NA>
## 5    <NA>     <NA>     <NA>
## 6    <NA>     <NA>     <NA>
## 7    <NA>     <NA>     <NA>
## 8    <NA>     <NA>     <NA>
## 9    <NA>     <NA>     <NA>
## 10              21       12
## 11   <NA>     <NA>     <NA>
## 12              21       37
## 13   <NA>     <NA>     <NA>
## 14              29       15
## 15   <NA>     <NA>     <NA>
## 16   <NA>     <NA>     <NA>
## 17              17       47
## 18   <NA>     <NA>     <NA>
## 19              26       48
## 20   <NA>     <NA>     <NA>
## 21     1:       02       45
## 22   <NA>     <NA>     <NA>
## 23   <NA>     <NA>     <NA>
## 24              18       57
## 25   <NA>     <NA>     <NA>
## 26              44       54
## 27   <NA>     <NA>     <NA>
## 28              55       19
## 29   <NA>     <NA>     <NA>
## 30   <NA>     <NA>     <NA>
## 31              26       14
## 32   <NA>     <NA>     <NA>
## 33              23       30
## 34   <NA>     <NA>     <NA>
## 35   <NA>     <NA>     <NA>
## 36   <NA>     <NA>     <NA>
## 37   <NA>     <NA>     <NA>
## 38              28       10
## 39   <NA>     <NA>     <NA>
## 40              43       15
## 41   <NA>     <NA>     <NA>
## 42              58       31
## 43   <NA>     <NA>     <NA>
## 44   <NA>     <NA>     <NA>
## 45              26       58
## 46   <NA>     <NA>     <NA>
## 47              39       41
## 48   <NA>     <NA>     <NA>
## 49              54       26
## 50   <NA>     <NA>     <NA>
## 51   <NA>     <NA>     <NA>
```

```
video_lengths <-
  video_lengths %>%
  mutate(hour_1 = str_replace_all(hour_1, ":", ""),
         hour_2 = str_replace_all(hour_2, ":", ""),
```

```
        hour_3 = str_replace_all(hour_3, ":", ""),
        hour_4 = str_replace_all(hour_4, ":", ""))

video_lengths <- data.frame(sapply(video_lengths, as.integer))

h <- c("hour_1", "hour_2", "hour_3", "hour_4")
m <- c("minute_1", "minute_2", "minute_3", "minute_4")
s <- c("second_1", "second_2", "second_3", "second_4")

video_lengths <-
  video_lengths %>%
  rowwise() %>%
  mutate(hours = sum(c_across(any_of(h)), na.rm = TRUE),
         minutes = sum(c_across(any_of(m)), na.rm = TRUE),
         seconds = sum(c_across(any_of(s)), na.rm = TRUE)) %>%
  ungroup() %>% select("hours", "minutes", "seconds")
```

```
channel_growth <- smosh %>% select(!(starts_with(("Video"))))
channel_growth["Total.Duration.of.Videos.Posted.That.Week"] <-
  video_lengths %>%
  mutate(minutes = floor(hours*60 + minutes + seconds/60)) %>% select(minutes)
```

```
channel_growth <- channel_growth[-c(1), ]
channel_growth["Viewer Growth"] <- as.numeric(growth[, 1])

channel_growth <- channel_growth %>%
  relocate("Viewer Growth", .after = "Total Viewers") %>% rename(
    "Total Views of Videos Posted that Week" = Total.Views.of.Videos.Posted.that.Week,
    "Number Videos Posted" = X..Videos.Posted,
    "Avg. Views of Videos Posted that Week" = Avg..Views.of.Videos.Posted.That.Week,
    "Total Duration of Videos Posted that Week" = Total.Duration.of.Videos.Posted.That.Week
)

rownames(channel_growth) <- NULL
channel_growth
```

```
##    Day     Month Year Total Viewers Viewer Growth
## 1    2   January 2023    2789909808       2898200
## 2   10   January 2023    2792792592       2882784
## 3   18   January 2023    2796442297       3649705
## 4   26   January 2023    2800576231       4133934
## 5    3  February 2023    2805812495       5236264
## 6   11  February 2023    2814945755       9133260
## 7   19  February 2023    2826026147      11080392
## 8   27  February 2023    2831436611       5410464
## 9    7     March 2023    2838461335       7024724
## 10  15     March 2023    2845410880       6949545
## 11  23     March 2023    2853837066       8426186
## 12  31     March 2023    2860609759       6772693
## 13   8     April 2023    2866469219       5859460
## 14  16     April 2023    2875708845       9239626
## 15  24     April 2023    2882705793       6996948
## 16   2       May 2023    2890092808       7387015
## 17  10       May 2023    2896647971       6555163
```

```
## 18  18       May 2023    2908538438      11890467
## 19  26       May 2023    2924547732      16009294
## 20   3      June 2023    2932733429       8185697
## 21  11      June 2023    2941948662       9215233
## 22  19      June 2023    2956435224      14486562
## 23  27      June 2023    2985892628      29457404
## 24   5      July 2023    3009881599      23988971
## 25  13      July 2023    3027823781      17942182
## 26  21      July 2023    3043670914      15847133
## 27  29      July 2023    3056263820      12592906
## 28   6    August 2023    3068262616      11998796
## 29  14    August 2023    3081768900      13506284
## 30  22    August 2023    3095929026      14160126
## 31  30    August 2023    3106367373      10438347
## 32   7 September 2023    3122512475      16145102
## 33  15 September 2023    3133129472      10616997
## 34  23 September 2023    3145011950      11882478
## 35   1   October 2023    3157591416      12579466
## 36   9   October 2023    3169191993      11600577
## 37  17   October 2023    3181076729      11884736
## 38  25   October 2023    3191044070       9967341
## 39   2  November 2023    3202661578      11617508
## 40  10  November 2023    3212486353       9824775
## 41  18  November 2023    3221934010       9447657
## 42  26  November 2023    3232103226      10169216
## 43   4  December 2023    3243182015      11078789
## 44  12  December 2023    3254301659      11119644
## 45  20  December 2023    3268840215      14538556
## 46  28  December 2023    3278211346       9371131
## 47   5   January 2024    3288597261      10385915
## 48  13   January 2024    3298588679       9991418
## 49  21   January 2024    3311328516      12739837
## 50  29   January 2024    3322953024      11624508
##    Total Views of Videos Posted that Week Number Videos Posted
## 1                                 1686685                    2
## 2                                 3868870                    3
## 3                                 3386381                    2
## 4                                 3519630                    3
## 5                                 2275954                    2
## 6                                 3436121                    2
## 7                                 1441645                    2
## 8                                 2620191                    3
## 9                                 6481835                    4
## 10                                3035189                    3
## 11                                4855481                    4
## 12                                1994275                    3
## 13                                4410841                    4
## 14                                4880999                    3
## 15                                2890042                    3
## 16                                6271837                    4
## 17                                3870685                    3
## 18                                7269584                    4
## 19                                2325618                    3
## 20                                5556022                    4
```

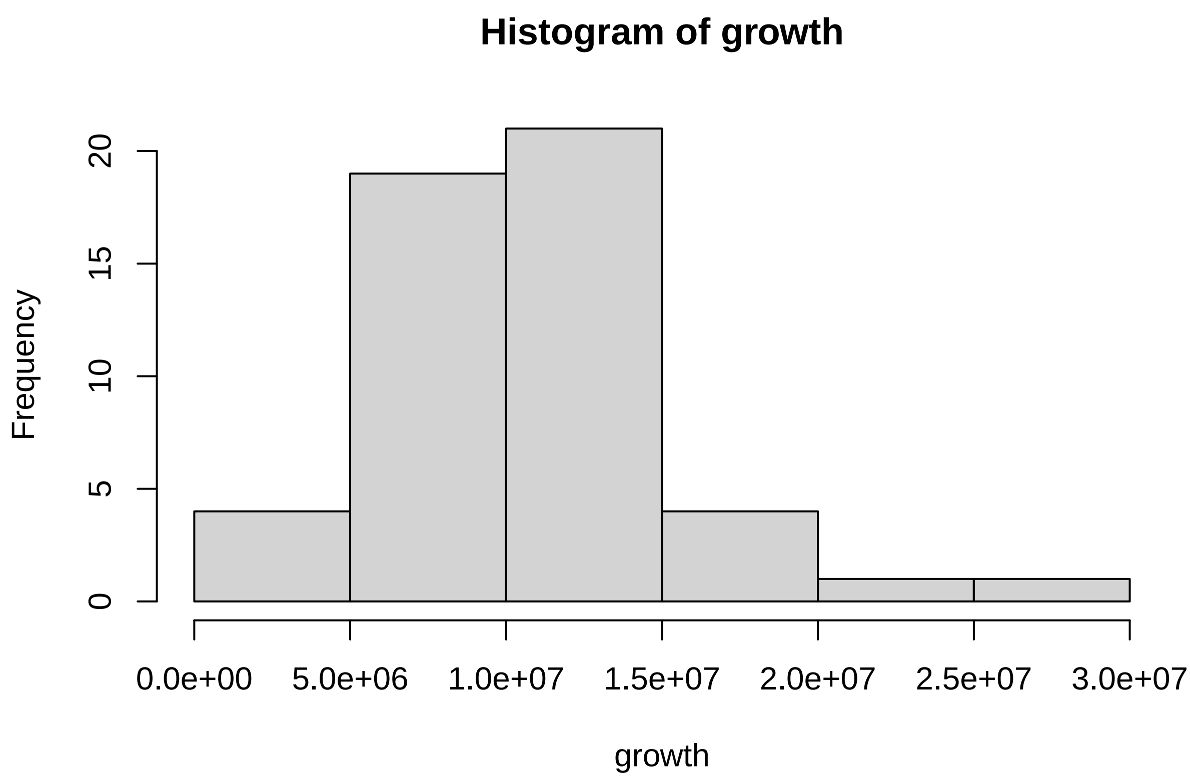
```
## 21                                  4252415                     3
## 22                                  5484814                     3
## 23                                 13882853                     4
## 24                                  4452081                     3
## 25                                  6999184                     4
## 26                                  3831462                     3
## 27                                  6313003                     4
## 28                                  4638467                     3
## 29                                  3403611                     3
## 30                                  6380212                     4
## 31                                  4726224                     3
## 32                                  7210340                     4
## 33                                  3661877                     3
## 34                                  5134042                     3
## 35                                  3129607                     3
## 36                                  4002739                     3
## 37                                  5739373                     4
## 38                                  2977481                     3
## 39                                  3629886                     4
## 40                                  2590855                     3
## 41                                  5377169                     4
## 42                                  2015374                     3
## 43                                  2186709                     2
## 44                                  3527577                     4
## 45                                  3100627                     3
## 46                                  3477939                     4
## 47                                  2102020                     3
## 48                                  2291395                     4
## 49                                  2913832                     3
## 50                                  2194338                     3
##    Avg. Views of Videos Posted that Week
## 1                                 843343
## 2                                1289623
## 3                                1693191
## 4                                1173210
## 5                                1137977
## 6                                1718061
## 7                                 720823
## 8                                 873397
## 9                                1620459
## 10                               1011730
## 11                               1213870
## 12                                664758
## 13                               1102710
## 14                               1627000
## 15                                963347
## 16                               1567959
## 17                               1290228
## 18                               1817396
## 19                                775206
## 20                               1389006
## 21                               1417472
## 22                               1828271
## 23                               3470713
```

```
## 24                              1484027
## 25                              1749796
## 26                              1277154
## 27                              1578251
## 28                              1546156
## 29                              1134537
## 30                              1595053
## 31                              1575408
## 32                              1802585
## 33                              1220626
## 34                              1711347
## 35                              1043202
## 36                              1334246
## 37                              1434843
## 38                               992494
## 39                               907472
## 40                               863618
## 41                              1344292
## 42                               671791
## 43                              1093355
## 44                               881894
## 45                              1033542
## 46                               869485
## 47                               700673
## 48                               572849
## 49                               971277
## 50                               731446
##    Total Duration of Videos Posted that Week
## 1                                        323
## 2                                        980
## 3                                         47
## 4                                         92
## 5                                         42
## 6                                         72
## 7                                         48
## 8                                         45
## 9                                        146
## 10                                        63
## 11                                       144
## 12                                        54
## 13                                       158
## 14                                       102
## 15                                        93
## 16                                       130
## 17                                       126
## 18                                       153
## 19                                        82
## 20                                       174
## 21                                       116
## 22                                       133
## 23                                       154
## 24                                       115
## 25                                       169
## 26                                       113
```

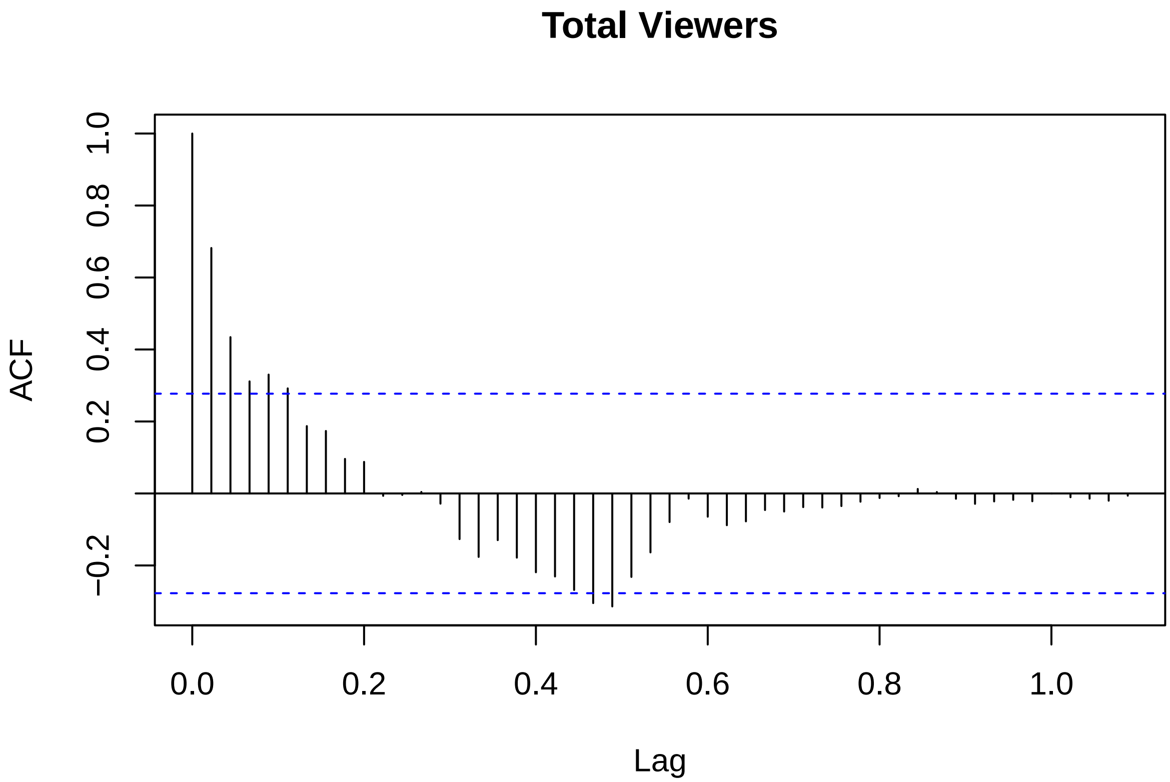
```
## 27                                              180
## 28                                              130
## 29                                              118
## 30                                              171
## 31                                              129
## 32                                              169
## 33                                              127
## 34                                              127
## 35                                              142
## 36                                              131
## 37                                              151
## 38                                              119
## 39                                              200
## 40                                              120
## 41                                              205
## 42                                              128
## 43                                              103
## 44                                              151
## 45                                              130
## 46                                              226
## 47                                              877
## 48                                             1023
## 49                                              125
## 50                                              158
```

First focus on viewership only

```r
hist(growth)
```

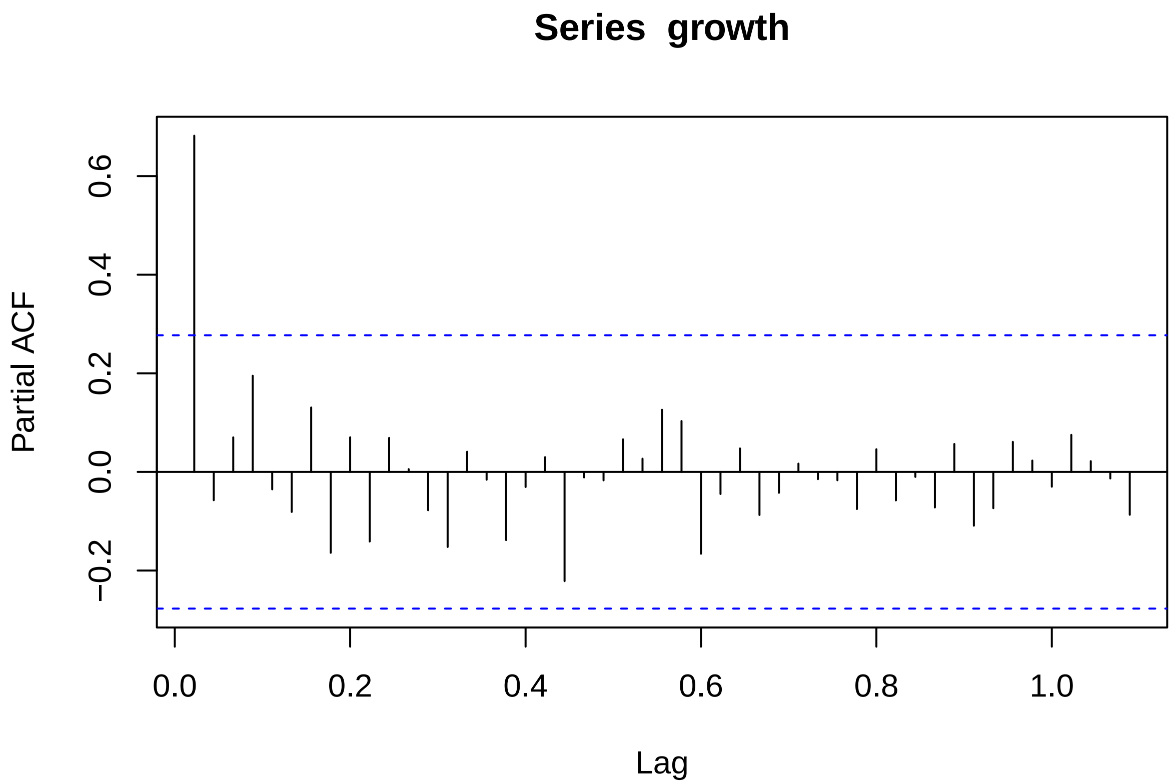


```r
acf(growth, lag.max = 50)
```




```r
# MA(2)?
```

```r
pacf(growth, lag.max = 50)
```




```
#only significant at lag 1 (AR(1))
```

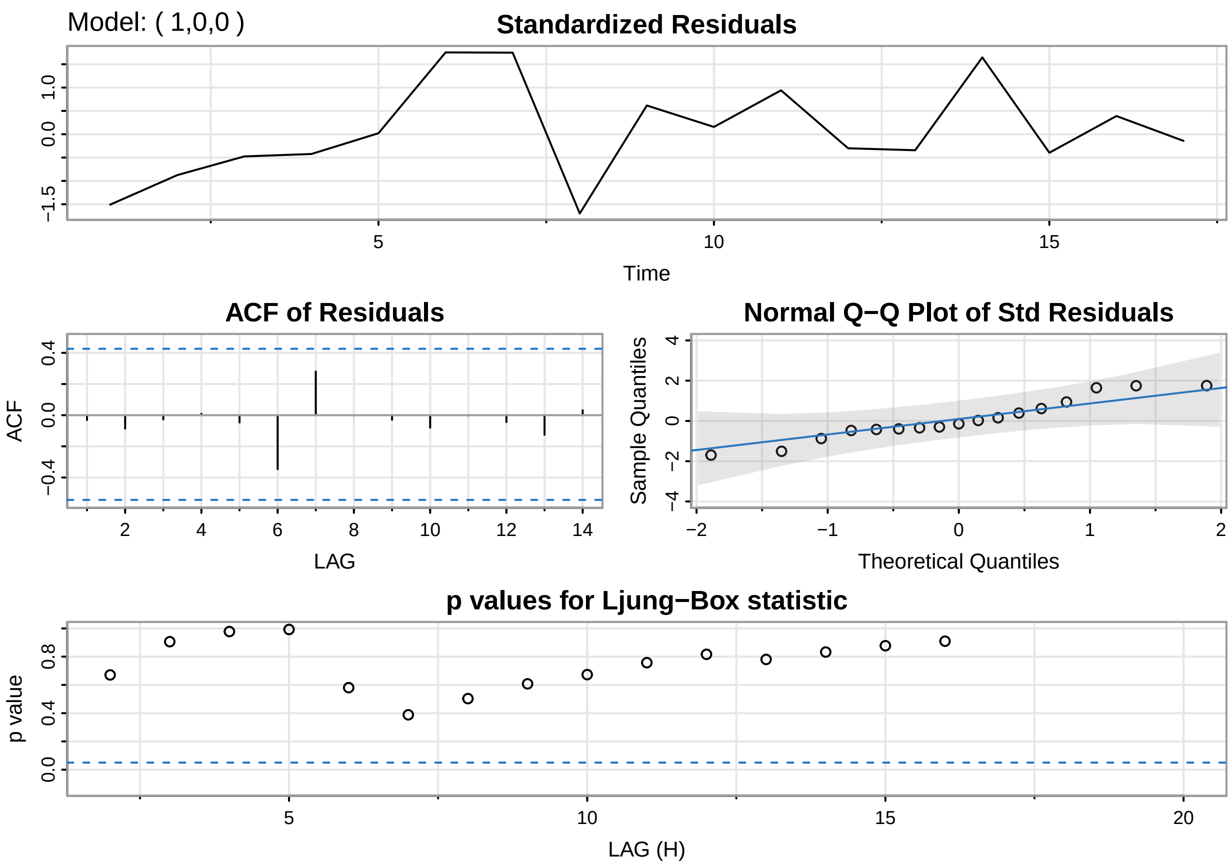
Forecasting growth without intervention

```
# train - test split
train_pre <- pre_roa[1:17]
test_pre <- pre_roa[18:22]
```

```
# AR(1) model

#t <- 1:length(train_pre)
#noint_model <- lm(formula = pre_roa ~ t)

ar1 <- sarima(train_pre, 1, 0, 0, P = 0, D = 0, Q = 0)
```

```
## initial  value 14.560831
## iter   2 value 14.454468
## iter   3 value 14.441005
## iter   4 value 14.415382
## iter   5 value 14.414876
## iter   6 value 14.414655
## iter   6 value 14.414655
## final  value 14.414655
## converged
## initial  value 14.496742
## iter   2 value 14.485651
## iter   3 value 14.481763
## iter   4 value 14.481496
## iter   5 value 14.481484
## iter   5 value 14.481484
```

```
## iter   5 value 14.481484
## final  value 14.481484
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##           Estimate          SE t.value p.value
## ar1         0.5026      0.2171  2.3152  0.0352
## xmean 6266169.1357 900995.8430  6.9547  0.0000
##
## sigma^2 estimated as 3.724061e+12 on 15 degrees of freedom
##
## AIC = 32.15379  AICc = 32.20421  BIC = 32.30082
##
```



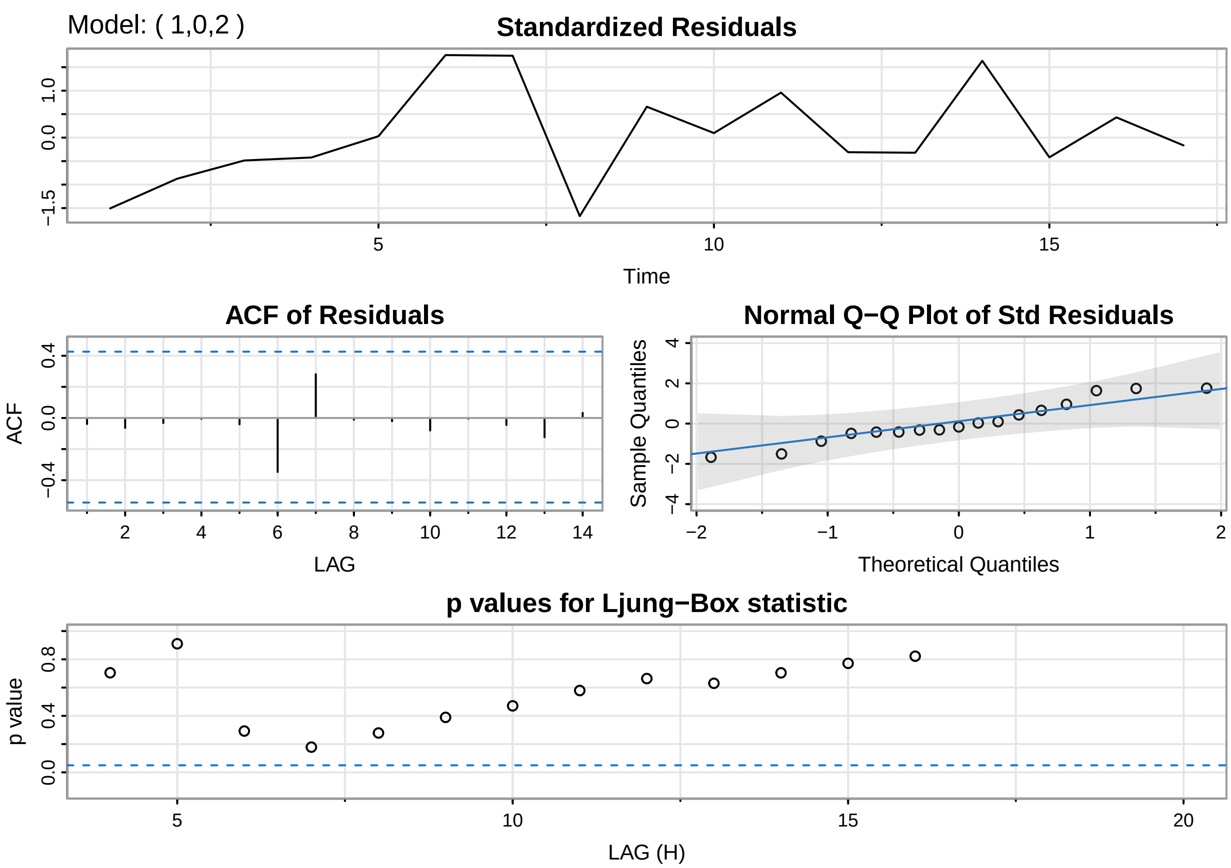

```r
# ARMA(1, 2)
ar1_ma2 <- sarima(train_pre, 1, 0, 2, P = 0, D = 0, Q = 0)
```

```
## initial  value 14.560831
## iter   2 value 14.549241
## iter   3 value 14.451523
## iter   4 value 14.419840
## iter   5 value 14.397602
## iter   6 value 14.280453
## iter   7 value 14.270940
## iter   8 value 14.240431
## iter   9 value 14.178564
```

```
## iter  10 value 14.164847
## iter  11 value 14.151673
## iter  12 value 14.110410
## iter  13 value 14.104645
## iter  14 value 14.082963
## iter  15 value 14.073591
## iter  16 value 14.069513
## iter  17 value 14.062022
## iter  18 value 14.057450
## iter  19 value 14.055355
## iter  20 value 14.053263
## iter  21 value 14.035690
## iter  22 value 14.025868
## iter  22 value 14.025868
## iter  23 value 14.025360
## iter  24 value 14.025280
## iter  25 value 14.025130
## iter  26 value 14.025016
## iter  27 value 14.024916
## iter  28 value 14.024796
## iter  29 value 14.024704
## iter  30 value 14.024584
## iter  31 value 14.024493
## iter  32 value 14.024374
## iter  33 value 14.024282
## iter  34 value 14.024164
## iter  35 value 14.024072
## iter  36 value 14.023955
## iter  37 value 14.023862
## iter  38 value 14.023746
## iter  39 value 14.023652
## iter  40 value 14.023538
## iter  41 value 14.023442
## iter  42 value 14.023330
## iter  43 value 14.023233
## iter  44 value 14.023122
## iter  45 value 14.023025
## iter  46 value 14.022915
## iter  47 value 14.022816
## iter  48 value 14.022708
## iter  49 value 14.022608
## iter  50 value 14.022501
## iter  51 value 14.022401
## iter  52 value 14.022295
## iter  53 value 14.022193
## iter  54 value 14.022089
## iter  55 value 14.021986
## iter  56 value 14.021883
## iter  57 value 14.021780
## iter  58 value 14.021678
## iter  59 value 14.021573
## iter  60 value 14.021473
## iter  61 value 14.021368
## iter  62 value 14.021268
```

```
## iter  63 value 14.021162
## iter  64 value 14.021064
## iter  65 value 14.020957
## iter  66 value 14.020860
## iter  67 value 14.020752
## iter  68 value 14.020657
## iter  69 value 14.020548
## iter  70 value 14.020453
## iter  71 value 14.020343
## iter  72 value 14.020251
## iter  73 value 14.020140
## iter  74 value 14.020048
## iter  75 value 14.019936
## iter  76 value 14.019846
## iter  77 value 14.019733
## iter  78 value 14.019644
## iter  79 value 14.019531
## iter  80 value 14.019443
## iter  81 value 14.019328
## iter  82 value 14.019242
## iter  83 value 14.019126
## iter  84 value 14.019042
## iter  85 value 14.018925
## iter  86 value 14.018841
## iter  87 value 14.018724
## iter  88 value 14.018641
## iter  89 value 14.018523
## iter  90 value 14.018442
## iter  91 value 14.018323
## iter  92 value 14.018243
## iter  93 value 14.018123
## iter  94 value 14.018044
## iter  95 value 14.017923
## iter  96 value 14.017846
## iter  97 value 14.017724
## iter  98 value 14.017648
## iter  99 value 14.017525
## iter 100 value 14.017450
## final  value 14.017450
## stopped after 100 iterations
## initial  value 14.612216
## iter   2 value 14.550963
## iter   3 value 14.487569
## iter   4 value 14.485066
## iter   5 value 14.483358
## iter   6 value 14.483209
## iter   7 value 14.483077
## iter   8 value 14.483011
## iter   9 value 14.482547
## iter  10 value 14.482000
## iter  11 value 14.481448
## iter  12 value 14.481219
## iter  13 value 14.481209
## iter  14 value 14.481167
```

```
## iter  15 value 14.481119
## iter  16 value 14.481089
## iter  17 value 14.481084
## iter  18 value 14.481083
## iter  18 value 14.481083
## iter  18 value 14.481083
## final  value 14.481083
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##           Estimate          SE t.value p.value
## ar1         0.5352      0.7279  0.7353  0.4752
## ma1        -0.0235      0.7456 -0.0316  0.9753
## ma2        -0.0378      0.3833 -0.0985  0.9230
## xmean 6262476.1788 931110.8309  6.7258  0.0000
##
## sigma^2 estimated as 3.720567e+12 on 13 degrees of freedom
##
## AIC = 32.38828  AICc = 32.58436  BIC = 32.63334
##
```



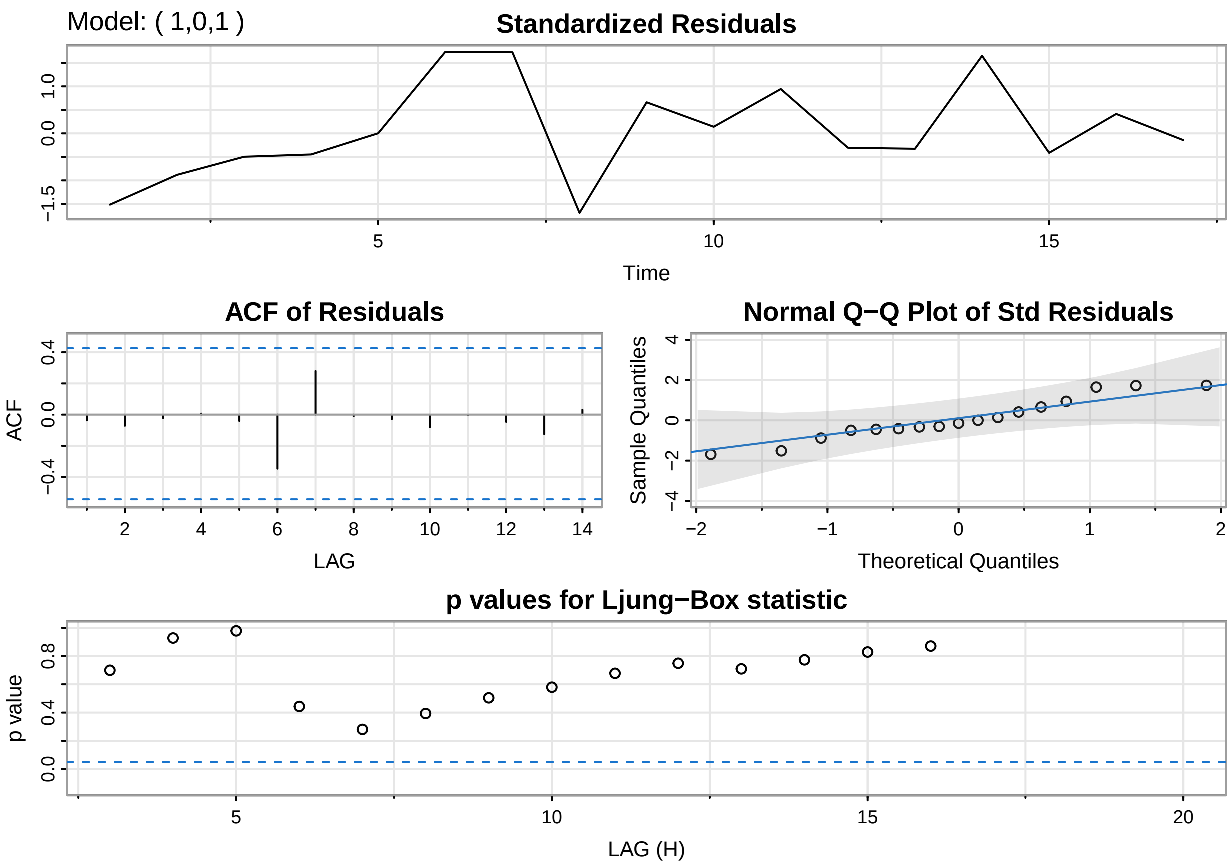

```
# ARMA(1, 1)
ar1_ma1 <- sarima(train_pre, 1, 0, 1, P = 0, D = 0, Q = 0)
```

```
## initial  value 14.560831
## iter   2 value 14.550320
```

```
## iter   3 value 14.443753
## iter   4 value 14.419103
## iter   5 value 14.404919
## iter   6 value 14.363494
## iter   7 value 14.353003
## iter   8 value 14.232056
## iter   9 value 14.220365
## iter  10 value 14.202507
## iter  11 value 14.193284
## iter  12 value 14.184163
## iter  13 value 14.183098
## iter  14 value 14.170050
## iter  15 value 14.163938
## iter  16 value 14.157764
## iter  17 value 14.147669
## iter  18 value 14.132436
## iter  19 value 14.117223
## iter  20 value 14.117007
## iter  21 value 14.107178
## iter  22 value 14.060340
## iter  23 value 14.059482
## iter  24 value 14.057704
## iter  25 value 14.056848
## iter  26 value 14.011620
## iter  26 value 14.011620
## iter  27 value 14.011006
## iter  27 value 14.011006
## iter  28 value 14.010965
## iter  29 value 14.010965
## iter  30 value 14.010916
## iter  31 value 14.010913
## iter  32 value 14.010866
## iter  33 value 14.010864
## iter  34 value 14.010817
## iter  35 value 14.010815
## iter  36 value 14.010768
## iter  37 value 14.010765
## iter  38 value 14.010718
## iter  39 value 14.010716
## iter  40 value 14.010669
## iter  41 value 14.010667
## iter  42 value 14.010620
## iter  43 value 14.010618
## iter  44 value 14.010571
## iter  45 value 14.010569
## iter  46 value 14.010522
## iter  47 value 14.010520
## iter  48 value 14.010473
## iter  49 value 14.010471
## iter  50 value 14.010424
## iter  51 value 14.010422
## iter  52 value 14.010375
## iter  53 value 14.010373
## iter  54 value 14.010327
```

```
## iter  55 value 14.010325
## iter  56 value 14.010278
## iter  57 value 14.010276
## iter  58 value 14.010230
## iter  59 value 14.010228
## iter  60 value 14.010181
## iter  61 value 14.010179
## iter  62 value 14.010133
## iter  63 value 14.010131
## iter  64 value 14.010085
## iter  65 value 14.010083
## iter  66 value 14.010036
## iter  67 value 14.010034
## iter  68 value 14.009988
## iter  69 value 14.009986
## iter  70 value 14.009940
## iter  71 value 14.009938
## iter  72 value 14.009892
## iter  73 value 14.009890
## iter  74 value 14.009844
## iter  75 value 14.009842
## iter  76 value 14.009796
## iter  77 value 14.009795
## iter  78 value 14.009749
## iter  79 value 14.009747
## iter  80 value 14.009701
## iter  81 value 14.009699
## iter  82 value 14.009653
## iter  83 value 14.009652
## iter  84 value 14.009606
## iter  85 value 14.009604
## iter  86 value 14.009558
## iter  87 value 14.009557
## iter  88 value 14.009511
## iter  89 value 14.009509
## iter  90 value 14.009464
## iter  91 value 14.009462
## iter  92 value 14.009416
## iter  93 value 14.009415
## iter  94 value 14.009369
## iter  95 value 14.009368
## iter  96 value 14.009322
## iter  97 value 14.009321
## iter  98 value 14.009275
## iter  99 value 14.009274
## iter 100 value 14.009228
## final  value 14.009228
## stopped after 100 iterations
## initial  value 14.612216
## iter   2 value 14.560760
## iter   3 value 14.484787
## iter   4 value 14.484389
## iter   5 value 14.483217
## iter   6 value 14.481384
```

```
## iter   7 value 14.481354
## iter   7 value 14.481353
## iter   7 value 14.481353
## final  value 14.481353
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##            Estimate          SE t.value p.value
## ar1          0.4734      0.4967  0.9532  0.3567
## ma1          0.0365      0.5478  0.0666  0.9478
## xmean 6277441.4932 898531.8875  6.9863  0.0000
##
## sigma^2 estimated as 3.723499e+12 on 14 degrees of freedom
##
## AIC = 32.27117  AICc = 32.37977  BIC = 32.46722
##
```




```r
# models virtually indistinguishable from eachother (AR(1) slightly
# lower AIC and BIC and has less parameters)
print(ar1$ICs)
```

```
##      AIC     AICc      BIC
## 32.15379 32.20421 32.30082
```

```r
print(ar1_ma2$ICs)
```

```
##      AIC     AICc      BIC
```

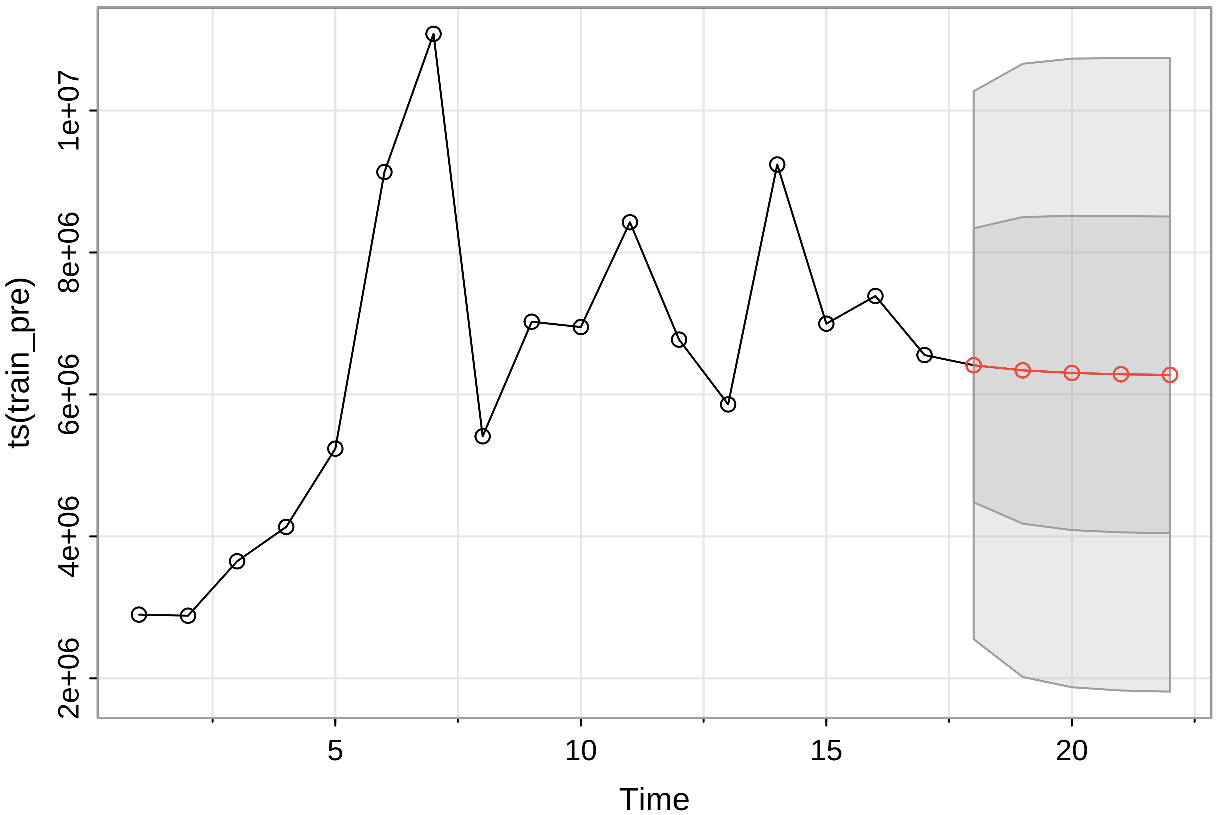
```
## 32.38828 32.58436 32.63334
```

```
print(ar1_ma1$ICs)
```

```
##      AIC     AICc      BIC
## 32.27117 32.37977 32.46722
```

```
# no pattern in residuals (no autocorrelation)
# standardized residuals over time still show some
```

```
# test model on test data
#ar1_forecast <- forecast(train_pre, h = 5)
ar1_forecast <- sarima.for(ts(train_pre), n.ahead = 5, 1, 0, 0)
```




```
ar1_pred <- ar1_forecast$pred
ar1_pred
```

```
## Time Series:
## Start = 18
## End = 22
## Frequency = 1
## [1] 6411432 6339185 6302871 6284617 6275442
```

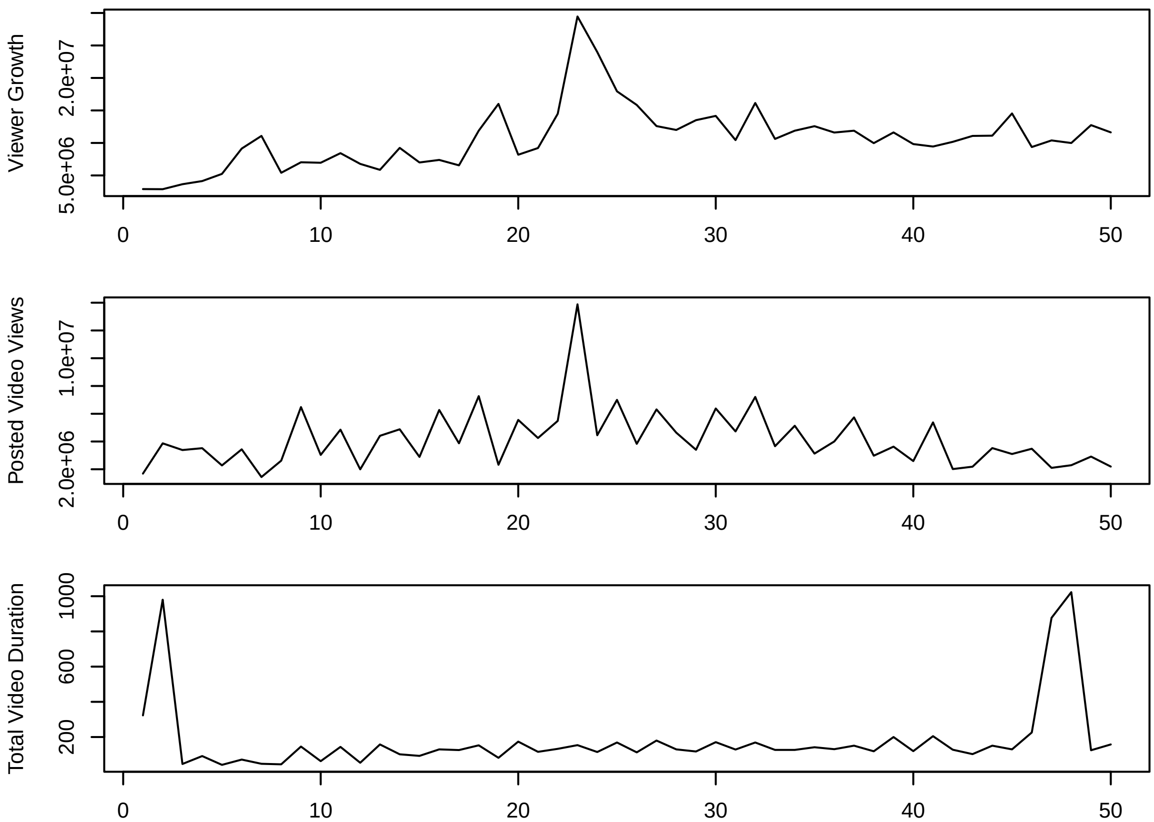
```
#unlist(list(pre_roa, noint_pred))

# predictions for test_pre
mape <- function(actual, prediction){
  return(mean(abs((actual - prediction)/actual)) * 100)
}
mape(test_pre, ar1_pred)
```

```
## [1] 43.59331
```

```r
# high mape makes sense - peak right after train set ends
```

Multivariate analysis

```r
par(mfrow = c(3, 1), mar = c(2, 6, 2, 2))
ts.plot(channel_growth["Viewer Growth"], xlab = "Time",
        ylab = "Viewer Growth")
ts.plot(channel_growth["Total Views of Videos Posted that Week"],
        xlab = "Time", ylab = "Posted Video Views")
ts.plot(channel_growth["Total Duration of Videos Posted that Week"],
        xlab = "Time", ylab = "Total Video Duration")
```



Sometimes, peaks in viewership do not coincide with that weeks video views and vice versa. Videos that may have performed well at the time may not have performed well in comparison to other weeks.

Videos posted during Anthony's return have maintained the highest overall views at the time and going foreward.

Video duration consistent aside from December and January when they posted 4-9 hour themed compilations of videos from that year.

Notes:

Viewers instead of subscribers - Smosh channels old and contained many kinds of content and cast members (don't have accurate data for those)

- Viewership based on number of new channel views in 8 day period

- Viewership from the time it was measured, but average video views recorded second week of January, so they include all views from video posted until January 2024.

8 day period instead of 7/1 week - this was based on the way videos were posted on the channel

- 45 8 day blocks (7 days inclusive)

Use viewership changes vs total views - total views don't capture changes in viewership well (ie. 200,000 is a big jump over 8 days, but very small compared to 3mil)

- no way to know who watched what videos (and different series/video types have different audiences)

Anthony usually makes appearences on Smosh main channel and Smosh Pit.

- Pit felt more interesting to follow overtime since all the cast appears there