Capstone

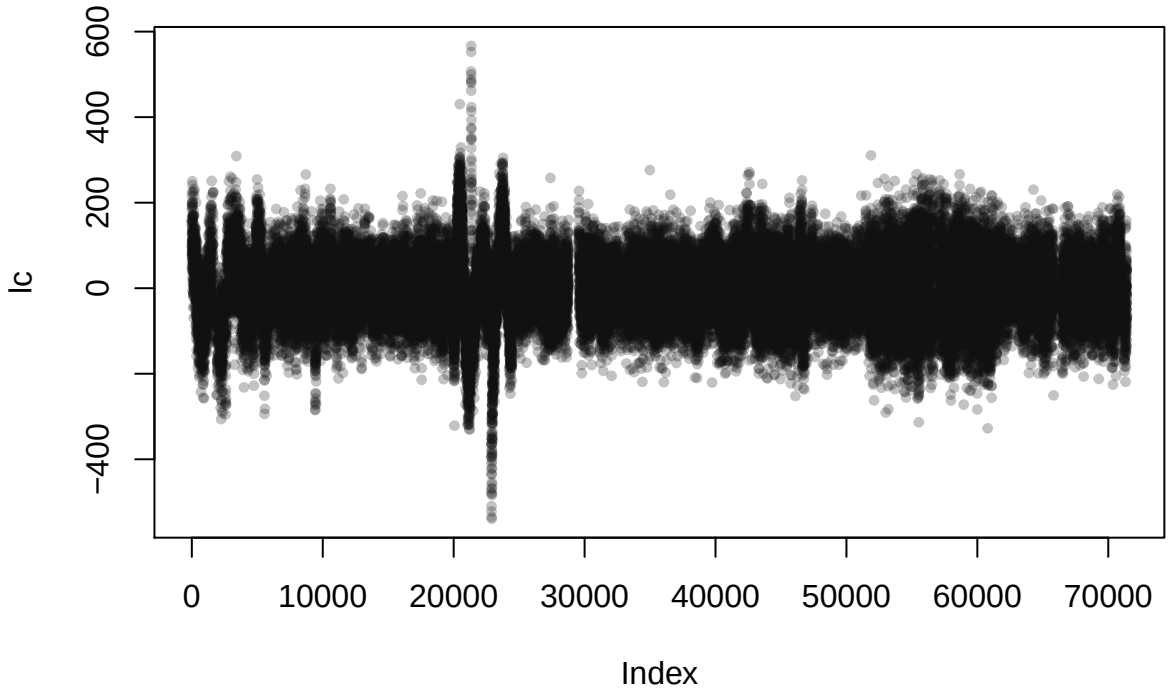
```
"The first step in the process is to subtract long-term dependencies in light-curve data
  by subtracting out a third-order polynomial that is fitted on the light curve"
```

```
## [1] "The first step in the process is to subtract long-term dependencies in light-curve data\n  by su
```

```
lc = load("000757076_results.Rdat")
lc = results$Lightcurve

summary(lc)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
## -539.112   -50.978    -1.478    -1.460    49.054   566.685     17729
```

```
#Plot the original, raw light curve
plot(lc, pch=20, cex=0.9, col = "#11111140")
```



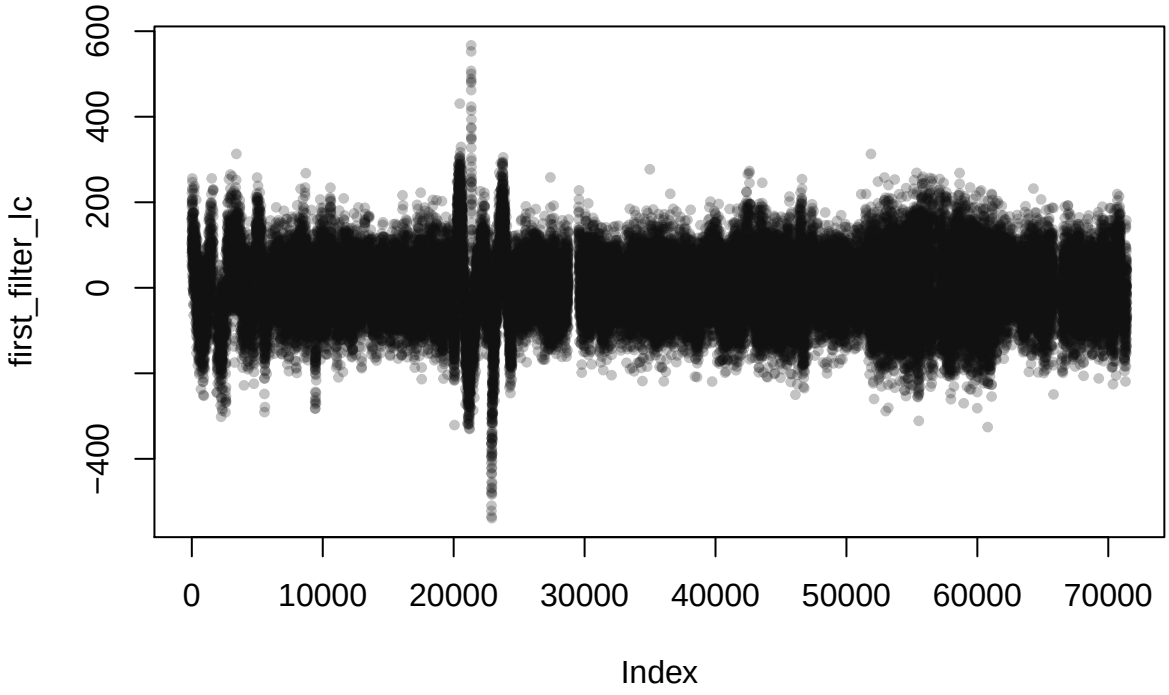
```r
evaluation_sequence = seq(1, length(lc))

#Fitting a third order polynomial on the raw light curve data
third_order_poly = lm(lc ~ poly(evaluation_sequence, 3), na.action = na.omit)
summary(third_order_poly)
```

```
##
## Call:
## lm(formula = lc ~ poly(evaluation_sequence, 3), na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -538.89  -49.57   -0.06   50.60  566.93
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -1.5212     0.3428  -4.437 9.12e-06 ***
## poly(evaluation_sequence, 3)1   60.6152    92.9089   0.652  0.51414
## poly(evaluation_sequence, 3)2 -133.6810    93.6005  -1.428  0.15324
## poly(evaluation_sequence, 3)3  262.8906    91.9140   2.860  0.00424 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79.31 on 53694 degrees of freedom
##   (17729 observations deleted due to missingness)
## Multiple R-squared:  0.0001891,  Adjusted R-squared:  0.0001332
## F-statistic: 3.385 on 3 and 53694 DF,  p-value: 0.0173
```

```r
sequence  = data.frame(x = seq(1, length(lc)))

predictions = predict(third_order_poly, newdata = sequence)

first_filter_lc = lc - predictions

#The first polynomial fitting shows a mean and median that are closer to 0
summary(first_filter_lc)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
## -538.893  -49.572   -0.064    0.000   50.604  566.930    17729
```

```r
plot(first_filter_lc, pch=20, cex=0.9, col = "#11111140")
```




```
"Light curve sequences must not contain gaps of more than 0.125 days,
  which translates to 6 NA's in a row. In order to be considered for
  analysis, a continuous sequence must also be at least 2 days in length"
```

```
## [1] "Light curve sequences must not contain gaps of more than 0.125 days, \n  which translates to 6 N
```

```
missing = 0

continuous_sequences = c()

start_of_sequences = c()
new = FALSE

"This for loop keeps track of how many NA's are in a row. If 6 are counted,
  then new is set to true and the start of the new sequence is recorded"
```

```
## [1] "This for loop keeps track of how many NA's are in a row. If 6 are counted,\n  then new is set to
```

```
for (i in 1:length(first_filter_lc)){

  if (!is.na(lc[i]) & new == TRUE){
    start_of_sequences = append(start_of_sequences, i)
  }
```

```
  if (is.na(lc[i])){
    missing = missing + 1
  }

  else{
    missing = 0
    new = FALSE
  }

  if (missing == 6){
    new = TRUE
  }
}

end_of_sequences = c()
missing = 0

for (i in 1:length(first_filter_lc)){

  if (is.na(lc[i])){
    missing = missing + 1
  }

  else{
    missing = 0
  }

  if (missing == 6){
    end_of_sequences = append(end_of_sequences, i)
  }

}



for (j in 1:length(start_of_sequences)){
  if (j < length(end_of_sequences)-1 & end_of_sequences[j+1]-start_of_sequences[j]>=96){
    continuous_sequences = append(continuous_sequences, j)
  }

}
```

```
"This step is very similar to the first one, however the fitting and subtraction of the polynomial
  is done within each continuous sequence of the light curve"
```

```
## [1] "This step is very similar to the first one, however the fitting and subtraction of the polynomia
```

```
#So as to not overwrite the first step's progress
second_filter_lc = first_filter_lc

for (i in 1:length(continuous_sequences)){

  continuous_segment  = second_filter_lc[start_of_sequences[i]:end_of_sequences[i+1]]
```

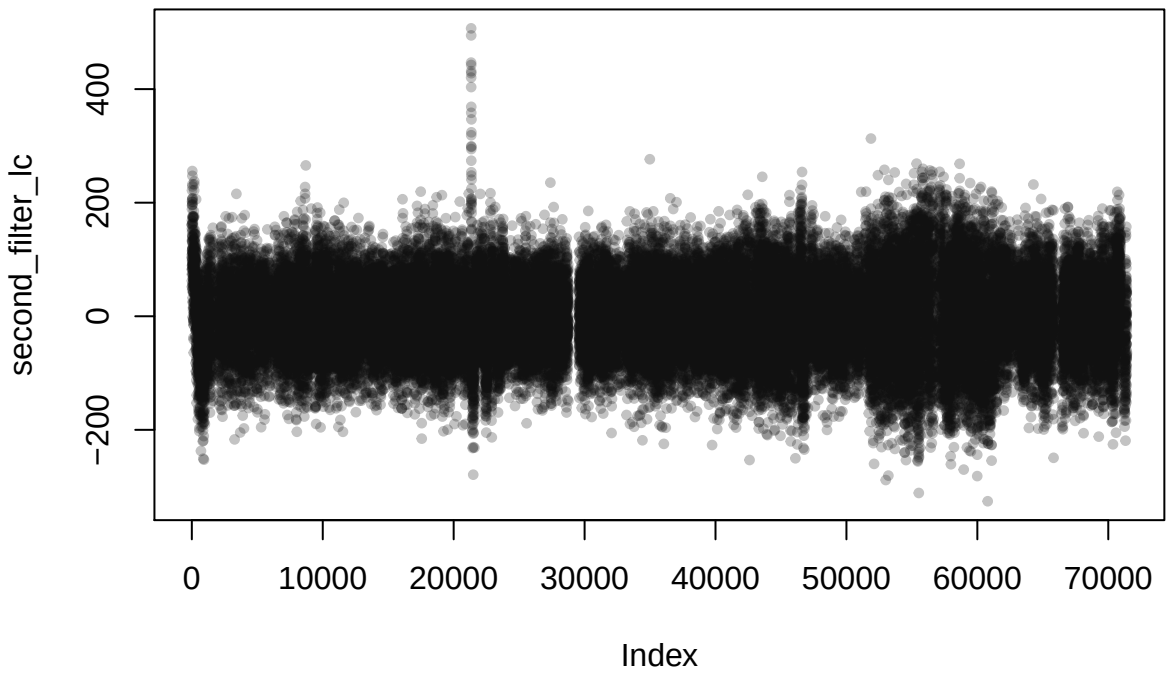
```
  eval_sequence = seq(1, length(continuous_segment))

  third_order_poly = lm(continuous_segment ~ poly(eval_sequence, 3, raw = TRUE))

  sequence  = data.frame(x = seq(1, length(continuous_segment)))
  predictions = predict(third_order_poly, newdata = sequence)

  second_filter_lc[start_of_sequences[i]:end_of_sequences[i+1]] = lc[start_of_sequences[i]:end_of_sequer
}
```

```
## Warning in predict.lm(third_order_poly, newdata = sequence): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(third_order_poly, newdata = sequence): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(third_order_poly, newdata = sequence): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(third_order_poly, newdata = sequence): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(third_order_poly, newdata = sequence): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(third_order_poly, newdata = sequence): prediction from a
## rank-deficient fit may be misleading

## Warning in predict.lm(third_order_poly, newdata = sequence): prediction from a
## rank-deficient fit may be misleading
```

```
summary(second_filter_lc)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
## -325.759  -46.238   -1.384   -0.935   44.998  507.022    17729
```

```
#It can be seen visually that the values are within a narrower range
plot(second_filter_lc, pch=20, cex=0.9, col = "#11111140")
```


second_filter_lc
400
200
0
−200
0
10000
20000
30000
40000
50000
60000
70000
Index