FPS assignment

Rajdeep Brahma

November 27 2020

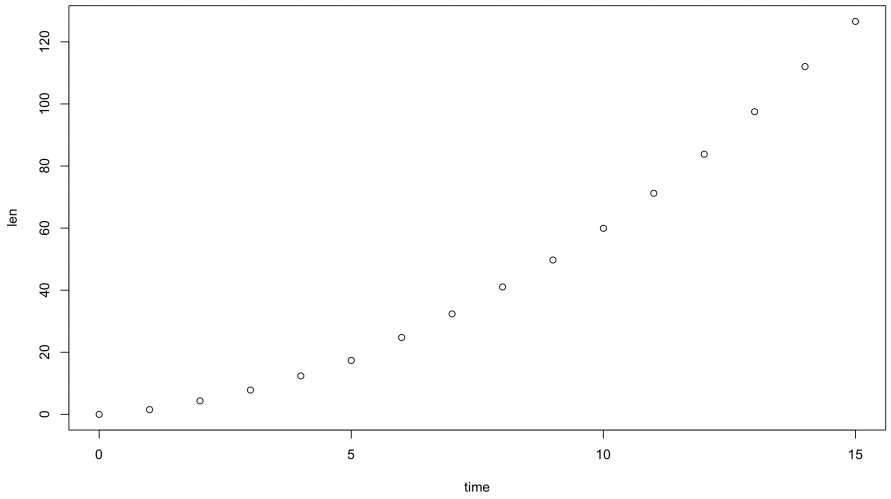
AIM:

In this project I want to calculate FPS of my mobile phone camera by using the common physics law for a freely falling object i.e $s = ut + 0.5 * g * t^2$

Methodology:

I dropped a red ball along the wall of my bed room and took a video of the entire process.Then using VLC I extracted the frames from my video and later from the still frames I found out the pixel co ordinate.Assuming we know the value of g quite correctly (980.6 cm/s^2) we will find a 95 % confidence interval estimate of FPS.

Description of Data



In the above plot x axis is time measured in frames and the y axis is the pixel co ordinate measured in cm after suitable conversion.(1 pixel co ordinate = $\frac{15.2}{70}$ cm).As expected from the formula $s = ut + 0.5gt^2$ the graph is a parabola.I deliberately set s=0 when t=0.

Model:

First we will try out the model $s = ut + 0.5gt^2$ which we know is correct by laws of physics.

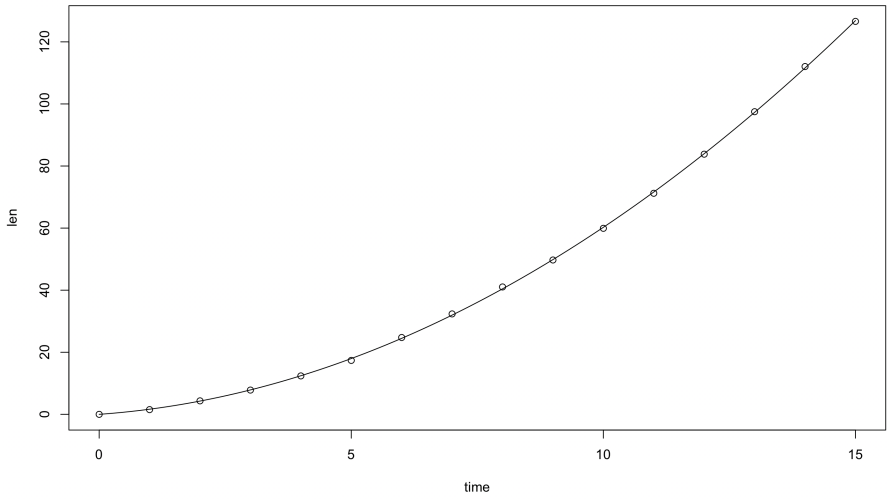
```
Call:
lm(formula = len ~ I(time^2) + I(time) - 1)

Coefficients:
I(time^2)    I(time)
   0.4862     1.1639
```

Now we will see if coefficient of t is significant or not since I thought it should not be significant as I released the ball from rest.

```
Analysis of Variance Table

Model 1: len ~ I(time^2) - 1
Model 2: len ~ I(time^2) + I(time) - 1
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1     15 106.051
2     14   1.618  1    104.43 903.47 4.063e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hence we see from here that the coefficient of t is quite significant and we can not drop it,the reason of which I do not know.
So we will work with our initial model.

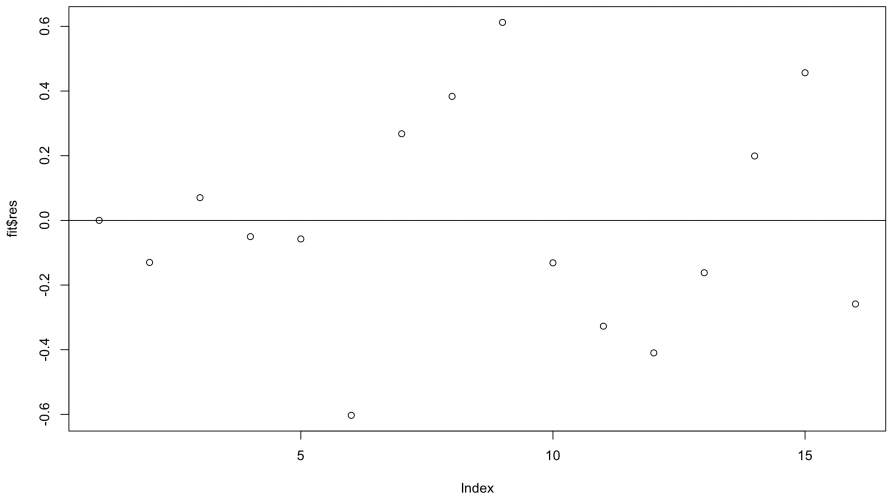
Diagnosis:



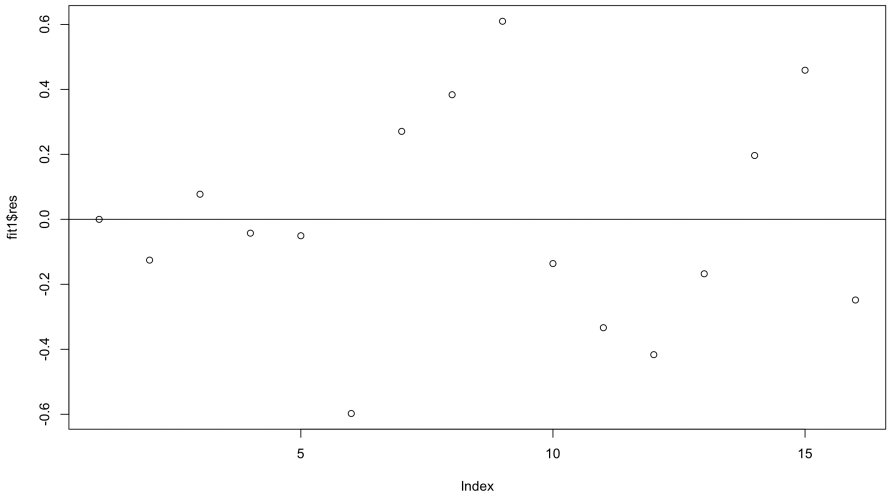
This is the fitted line which more or less fits the given data perfectly.Below I will attach the summary of the fit,r squared of whose looks pretty decent.

```
Call:
lm(formula = len ~ I(time^2) + I(time) - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-0.6029 -0.1861 -0.0539  0.2163  0.6123

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
I(time^2) 0.486199   0.003229  150.57  < 2e-16 ***
I(time)   1.163875   0.038721   30.06 4.06e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.34 on 14 degrees of freedom
Multiple R-squared:      1,  Adjusted R-squared:      1
F-statistic: 2.601e+05 on 2 and 14 DF,  p-value: < 2.2e-16
```

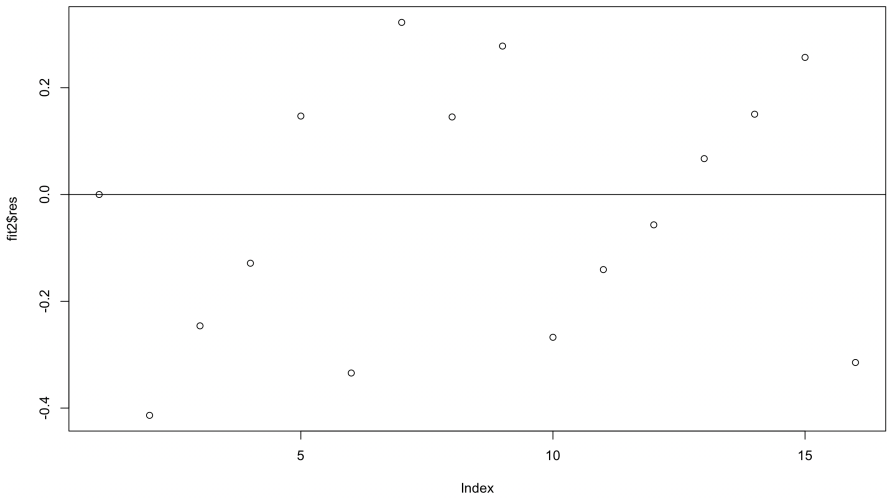
Now we will look at the error plot of our fitted model.



The errors are scattered around y=0 line.It seems that the plot has some sort of sinusoidal trend.So I will try to fit a higher order model in t(cubic) and look at the residual plot to see if it improves or not.



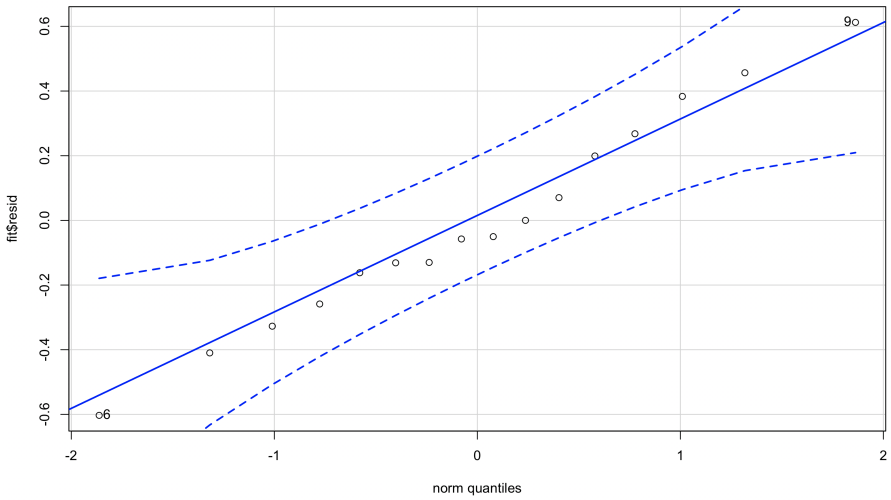
This plot is almost same as the previous one and has the same problem.
Now we will try out a weird fit $s = ut + 0.5gt^2 + csin(t)$.The intiution of such a fit is due to the sinusoidal trend in the residual plot.



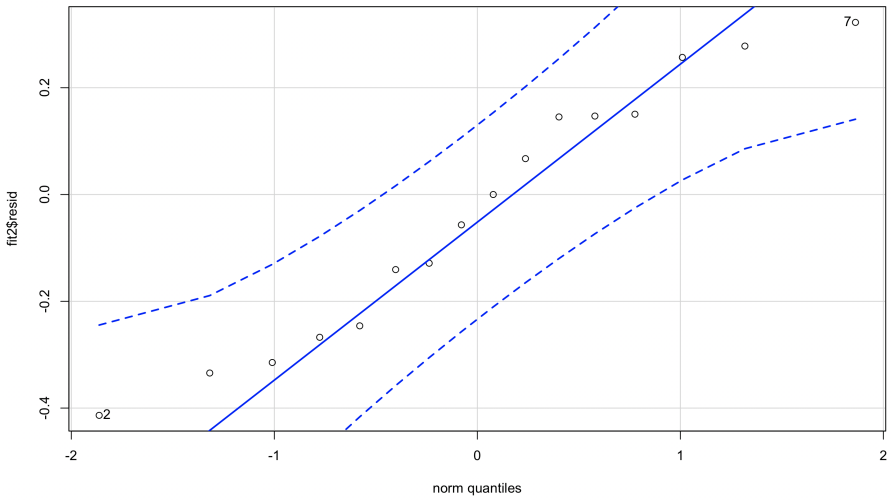
Now this plot is also not perfect but the sinusoidal trend seems to have decreased a bit.We will take a look at the coefficients of regression in this model.

```
Call:
lm(formula = len ~ I(time^2) + I(time) + I(sin(time)) - 1)

Coefficients:
   I(time^2)       I(time)  I(sin(time))
      0.4844        1.1803        0.3195
```

Now the coefficient of t^2 is 0.4844 here and 0.4862 in our model that we got from physics.Since this model is not guided by any physics law and the coefficient of t^2 is almost the same in both the models we will stick to our original model although AIC value of our model with sin term is -40.5 which is significantly lower than our original fit which has AIC value -32.7.
Now we will do QQ plots of the residuals



This is the QQ plot for $s = ut + 0.5ft^2$



This is the QQ plot for $s = ut + 0.5ft^2 + c(sint)$
We also did an influence diagnosis of our data.

```
Influence measures of
     lm(formula = len ~ I(time^2) + I(time) - 1) :

    dfb.I..2  dfb.I.t.     dffit cov.r   cook.d      hat inf
1  -2.24e-29  2.32e-29  2.32e-29 1.160 2.89e-58 2.77e-32
2   3.78e-02 -3.92e-02 -3.92e-02 1.148 8.20e-04 1.10e-02
3  -3.78e-02  3.96e-02  3.96e-02 1.196 8.42e-04 3.66e-02
4   3.75e-02 -3.96e-02 -3.97e-02 1.240 8.48e-04 6.75e-02
5   5.23e-02 -5.57e-02 -5.62e-02 1.278 1.69e-03 9.65e-02
6   7.05e-01 -7.63e-01 -7.74e-01 0.731 2.41e-01 1.19e-01
7  -2.87e-01  3.16e-01  3.25e-01 1.202 5.40e-02 1.31e-01
8  -4.07e-01  4.59e-01  4.84e-01 1.073 1.13e-01 1.34e-01
9  -6.37e-01  7.48e-01  8.26e-01 0.717 2.70e-01 1.27e-01
10  9.45e-02 -1.18e-01 -1.44e-01 1.279 1.10e-02 1.15e-01
11  1.65e-01 -2.36e-01 -3.47e-01 1.110 6.01e-02 1.04e-01
12  8.80e-02 -1.92e-01 -4.38e-01 1.010 9.15e-02 1.02e-01
13 -2.32e-02 -2.22e-02 -1.81e-01 1.267 1.72e-02 1.18e-01
14  1.18e-01 -5.07e-02  2.79e-01 1.310 4.08e-02 1.66e-01
15  6.10e-01 -4.00e-01  9.77e-01 1.068 4.25e-01 2.59e-01
16 -6.26e-01  4.69e-01 -8.35e-01 1.711 3.49e-01 4.14e-01   *
```

We found out the last data point to be influential which is sort of expected due to the blur at that position.

Calculation of FPS

We will now calculate the estimate of FPS from our fit.
y=0.5*g*t^2 where t is measured is second
y=b*t^2 where t is measured in frames
Say 1 sec = x frames
By simple calculations we can obtain $x^2 = \frac{g}{2b}$.
Value of x is 31.75692(31.69105 if we remove the last point as it is influential) in our model which is quite absurd since I expected it to be not more than 30.

Finding out 95% C.I for estimated FPS

We will do bootstrap on the errors 1000 times to get an estimate of mean and standard deviation of our FPS estimate.
The standard deviation was 0.09665696 and the mean was 31.75684.
So our 95 % C.I will be (mean – 1.96*sd, mean + 1.96*sd)=(31.56739,31.94628) assuming that our error follows Normal distribution with mean 0.

Success and Limitations of the Project:

The fitted line almost matched our given data.The two main problems I faced was the coefficient of t not being negligible and the error plot having a sinusoidal trend.The later one was sort of fixed by adding a sin(t) term but the reason of which I did not get by laws of physics.So I progressed with my original model although statistically the one with sin term was a better fit.The first problem is something which I could not figure out.Some possible sources of experimental error can be initial velocity of the ball not being zero,the time gap between two frames are not same,when the ball is blurred in later frames I took the co ordinate of the midpoint which may not be the best thing to do,etc.I think the mysterious coefficient of t is the cause of unexpected rise of my FPS estimate(since if there is no positive t term in my model then coefficient of t^2 i.e b should have been higher bringing down the value of my estimate of FPS) which should have been around 30.Also finally that the errors are homoscedastic may not be a proper assumption.This is because I always took the midpoint of the blur as my guess and that is most probably not a right guess as the speed of ball increases with time.Except these points the project overall went successfully and we got our **95 % C.I of estimated FPS as (31.56739,31.94628)**.