Monte-Carlo Simulation Report

Intro to R for Data Science

By

Anoop Nair

R00223644

Declaration

I hereby certify that this material which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent, that such work has been cited and acknowledged within the text of my work. I understand that my project documentation may be stored in the library at MTU, and may be referenced by others in the future.

Synopsis

In this report we discuss about the Monte Carlo simulation and how this approach is used to simulate the data and run linear predictive models on top of it. The use case we are addressing here is, when the predictive variable data is not known but we have the distribution of residuals (i.e., in the case of normal distribution we will be given the mean and the standard deviation) that it follows and the linear equation it adheres to, then we can simulate the data using this and then create predictive model on top of it. The assumption is the predictions or model created on simulated data closely resembles what we get from the real data.

What is Monte Carlo Simulation?

Monte Carlo simulation or the Monte Carlo method, considers all possible outcomes through randomization including the actual probabilities each will occur. It helps us in predictions in the absence of data, quantitatively assess the impact of risk, enable better forecasting, leading to better decision making under uncertainity. [reference https://www.palisade.com/monte-carlo-simulation/]

Our Use case:

We are given a dataset that captures the details of how CO2 emissions of a vehicle can vary depending on the different features of each vehicle. The sample is taken from a Canada Government official open data website.

So, I tried to use Monte Carlo simulation to simulate the values of Co2 emissions based on the residual mean and standard deviation for each model and checked How well the simulated data performed wrt to the linear models on the actual data.

Steps followed during simulations:

1) Fit 3 linear models on existing dataset; A reduced best fit model (as already provided) and 2 other models.
2) For each model get the Standard residual error, which is nothing but the standard deviation of the residuals (difference in original and predicted values).
3) Assuming that there is higher chance of residuals following a normal distribution, we generate a standard normal distribution with mean as zero and standard deviation, that of residuals.
4) Simulate n values of residuals following the distribution of point 3, where n is the number of records in the dataset.
5) Calculate the values of new CO2 using the linear formula of the specific model and the simulated residuals.
6) Fit a new model using the values of newly simulated CO2 emission values, to get the parameters of linear model created on simulated CO2 emission value.
7) Repeat the steps from 2 to 6 for the other 2 models.
8) Compare each model with eachother.
9) Also compare each of real model to their respective simulated model.

Explanation

At first, we fit 3 models one amongst them is the reduced best fit model. Below figure shows the model performance of the models. We can clearly see that Best fit model performs the best with Lowest RSE and highest multiple R-squared

	Best Fit Model (Model1)	Model 2	Model 3
Predictors	Enginesize,Fuel Consumption, Milespergallon_both,Cylinder	Fuelconsumption_both	Enginesize
RSE	18.25	23.2	30.72
Multiple R-squared	0.9028	0.8428	0.7244

Fig1: Comparing Model performance using original Co2 emissions.

Different Models

Model 1 (Best Fit Model):

Co2emmissions (g) = 224.20695+ 4.91*enginesize+ 6.91*Cylinder+ 5.67*Fuel_Consum_both -3.28*Milespergallon + Epsilon

First step was to build the Best first Linear Model. Then using the assumption that residuals are normal, we simulate the residuals using the distribution with mean as 0 and standard deviation as Residual standard Error from model and find n(total values in dataset) values of RSE using the rnorm function in R **(Check appendix for code)**

```
residuals_SD1=summary(lm_model1)$sigma.
residualSD1_simulateddata=rnorm(n=nrow(Co2data_Copy),mean=0,sd=residuals_SD1)
```

Later we use the values of residuals and the linear model to find the simulated values of CO2 emissions.

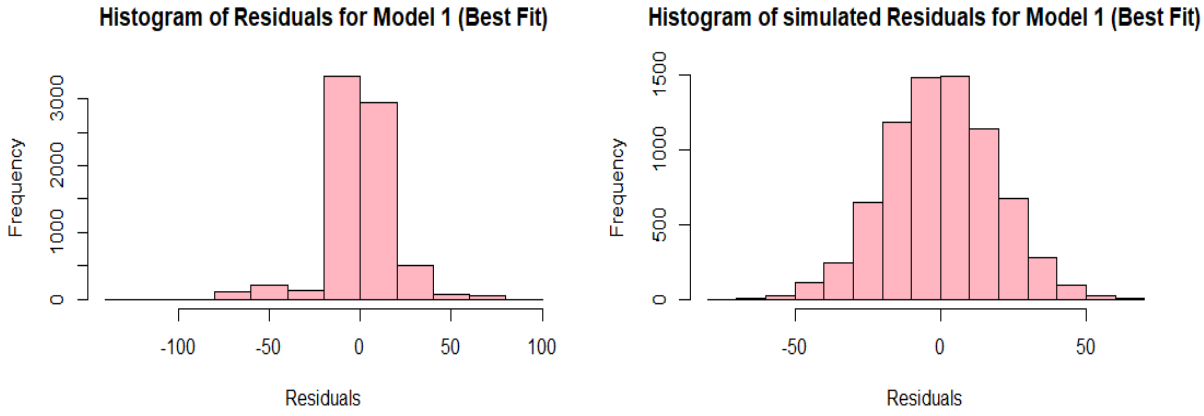

Fig2: Distribution difference between real and simulated residuals for Best Fit Model.

The simulated residuals look more normal than the original one as the rnorm function takes care that the residuals are well distributed and follow a bell curve. Average of real Co2 emissions is 250.5847 and simulated is 250.6337.

The median is also 246 & 247 respectively. Hence simulation doesn't change the central tendency and the spread of the Co2 emissions.

Let's look at the plot of residuals of other two models.

Model2 : Co2emmissions (g) = 46.76+ 18.57*Fuel_Consum_both + Epsilon

Model 3 : Co2emmissions (g) = 134.36 + 36.7 * Enginesize + Epsilon

Fig3: Distribution difference between real and simulated residuals for Model 2.

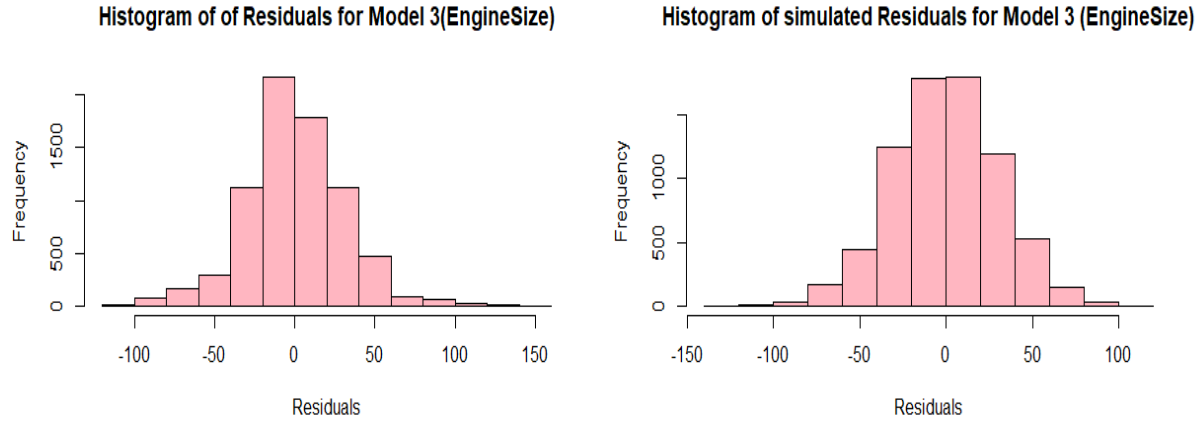

Fig4: Distribution difference between real and simulated residuals for Model 3.

From the residual plots between the 3 models, we can see that, Model1 has a lower residual spread. When we compare the results of the simulated model for all the 3 models, we get the below result.

Using Simulated Model (simulated Co2 values)			
	Best Fit Model (Model1)	**Model 2**	**Model 3**
Predictors	Enginesize,Fuel Consumption, Milespergallon_both,Cylinder	Fuelconsumption_both	Enginesize
RSE	18.76	23.1	30.57
Multiple R-squared	0.8978	0.845	0.729

Fig5: Comparing Model performance using simulated CO2 emissions.

By looking at Fig 5 we can see that Best Fit Model still is the Model1 (reduced best fitted model) using the simulated Co2 emission values. Also, when we compare Figure 5 on simulated data to the corresponding metrics on real data on Fig 1. We see that the Metrics on the whole doesn't change much for individual models. Hence, it is fair to say that the simulated data on the whole does average out well and provide the approximately the same result as the real data.

When we compare the coeffecients of the parameters and their standard errors of both the Model on real data and the simulated data for best fit model we get the below results.

Model 1 (Best Fit Model)				
Variable	**Coeffecient RealModel**	**Coeff Simulatedmodel**	**Standarderror RealModel**	**Standarderror Simulatedmodel**
Intercept	224.206953	224.607991	4.2046689	4.18501736
Enginesize	4.911002	5.747885	0.45801341	0.45587277
Cylinder	6.911625	6.306298	0.31290232	0.3114399
Fuelconsumption_both	5.679959	5.71893	0.22029303	0.21926344
Milespergallon_both	-3.285403	-3.286327	0.07760747	0.07724475

Fig6: Comparing Coeffecients & Standard Error of Real and simulated model.

We see that the coefficient and the Standard error are comparable and approximately the same for both Real and simulated Linear Models.

Conclusion

The Monte carlo simulation did a great job in predicting the data and maintaining the parameters of models using the real data. Hence it is safe to say that we can get a decent average prediction of the CO2 values by knowing just the Linear model and the distribution of the residuals. The more the number of simulations, lesser the variance and the data on the whole will resemble that of the real population data. Hence the parameters and the coeffecients will be similar to that of the real data. In the current scenario the reduced model performed well with both the real values of Co2 emissions and the simulated values. The Coeffecients and the standard errors of these parameters also doesn't change much for simulated Linear Model.

Appendix (code of the Montecarlo approach in order, also available in attached MonteCarlo.R script)

```
################# Monte Carlo Simulation ################

Co2data_Copy=Co2data ### copying the original dataset to Co2data
str(Co2data_Copy)
############ best fit model #########

lm_model1=lm(Co2emmissions~Enginesize+Cylinder+Fuelconsumption_both+Milespergallon_both,Co2data_Copy)
summary(lm_model1)

Coeff1=summary(lm_model1)$coefficients[1, 1]
Coeff1_engsize=summary(lm_model1)$coefficients[2, 1]
Coeff1_cyclinder=summary(lm_model1)$coefficients[3, 1]
Coeff1_Fuelconsumption_both=summary(lm_model1)$coefficients[4, 1]
Coeff1_Milespergallon_both=summary(lm_model1)$coefficients[5, 1]

residuals_SD1=summary(lm_model1)$sigma # 18.84

#### to simulate nrow values of residuals with mean as 0 and standard deviation from the model
set.seed(100)

residualSD1_simulateddata=rnorm(n=nrow(Co2data_Copy),mean=0,sd=residuals_SD1)### montecarlo simulation for residuals

sum(residualSD1_simulateddata) # 314.4737

######### calculating the New CO2 values using Best firt linear model and the simulated residuals

for (i in c(1:nrow(Co2data_Copy)))
{
  # iterating through linear model and residuals to calculate CO2emissions value
    Co2data_Copy[i,c('co2emission_model1')]=Coeff1+(Coeff1_engsize*Co2data_Copy[i,c('Enginesize')])+
      (Coeff1_cyclinder*Co2data_Copy[i,c('Cylinder')])+
    (Coeff1_Fuelconsumption_both*Co2data_Copy[i,c('Fuelconsumption_both')])+
    (Coeff1_Milespergallon_both*Co2data_Copy[i,c('Milespergallon_both')])+residualSD1_simulateddata[i]
}

Co2data_Copy['co2emission_model1']=round(Co2data_Copy['co2emission_model1'])
```

```r
Co2data_Copy['co2emission_model1']=round(Co2data_Copy['co2emission_model1'])

#### fit the best fit model using the simulated residuals ####

lm_model1_sim=lm(co2emission_model1~Enginesize+Cylinder+Fuelconsumption_both+Milespergallon_both,Co2data_Copy)
summary(lm_model1_sim)

par(mfrow=c(1,2))

######  histogram comparing the original residuals and simulated residuals using the simulated model

hist(lm_model1$resid, main="Histogram of Residuals for Model 1 (Best Fit)",
     ylab="Frequency",xlab='Residuals',col="light pink")

hist(lm_model1_sim$resid, main="Histogram of simulated Residuals for Model 1 (Best Fit)",
     ylab="Frequency",xlab='Residuals',col="light pink")

####### comparing the coefficients and standard errors of original to simulated model #####

summary(lm_model1)

sqrt(diag(vcov(lm_model1)))
sqrt(diag(vcov(lm_model1_sim)))

summary(lm_model1)$coefficients[, 1]
summary(lm_model1_sim)$coefficients[, 1]


######### average of real co2 emissions and simulated co2 emissions ########

mean(Co2data_Copy$Co2emmissions) #250.5847
mean(Co2data_Copy$co2emission_model1) #250.6337

median(Co2data_Copy$Co2emmissions) #250.5847
median(Co2data_Copy$co2emission_model1) #250.6337
```

```r
median(Co2data_Copy$Co2emmissions) #250.5847
median(Co2data_Copy$co2emission_model1) #250.6337

################################### 2nd Model using Fuelconsumption_both as the only predictor variable ##########################

lm_model2=lm(Co2emmissions~Fuelconsumption_both,Co2data_Copy)
summary(lm_model2)

Coeff2=summary(lm_model2)$coefficients[1, 1]
Coeff2_Fuelconsumption_both=summary(lm_model2)$coefficients[2, 1]



# The residual standard error from the model is the standard deviation of the residuals.

residuals_SD2=summary(lm_model2)$sigma # 23.19957

######## test for normality of residuals #####
qqnorm(lm_model2$resid)
qqline(lm_model2$resid)

#### to simulate values of residuals with mean as 0 and variance from the model

nrow(Co2data_Copy)
set.seed(100)
residualSD2_simulateddata=rnorm(n=nrow(Co2data_Copy),mean=0,sd=residuals_SD2)

sum(residualSD2_simulateddata) #399.8185


######## populating new Co2emisson value using Model2 ######
for (i in c(1:nrow(Co2data_Copy)))
{
  # iterating through linear model and residuals to calculate CO2emissions value
  Co2data_Copy[i,c('co2emission_model2')]=Coeff2+(
    Coeff2_Fuelconsumption_both*Co2data_Copy[i,c('Fuelconsumption_both')])+residualSD2_simulateddata[i]
}

Co2data_Copy['co2emission_model2']=round(Co2data_Copy['co2emission_model2']) # rounding off the CO2 values to remove decimal
```

```r
Co2data_Copy['co2emission_model2']=round(Co2data_Copy['co2emission_model2']) # rounding off the CO2 values to remove decimal

######### fit the 2nd Model  using the simulated values for CO2 present in co2emission_model2 #####

lm_model2_sim=lm(co2emission_model2~Fuelconsumption_both,Co2data_Copy)
summary(lm_model2_sim)

###### residuals comparison between real and simulated using histogram

hist(lm_model2$resid, main="Histogram of Residuals for Model 2 (FuelConsumption)",
     ylab="Frequency",xlab='Residuals',col="light pink") #Histogram of Residuals for best fit


hist(lm_model2_sim$resid, main="Histogram of simulated Residuals for Model 2 (FuelConsumption)",
     ylab="Frequency",xlab='Residuals',col="light pink") #Histogram of Residuals for best fit


####### comparing the coefficients and standard errors of original to simulated model #####

summary(lm_model1)

#### to calculate the standard error of the parameters
sqrt(diag(vcov(lm_model2)))
sqrt(diag(vcov(lm_model2_sim)))

#### to calculate the coeffecients value
summary(lm_model2)$coefficients[, 1]
summary(lm_model2_sim)$coefficients[, 1]

mean(Co2data_Copy$Co2emmissions) #250.5847
mean(Co2data_Copy$co2emission_model2) #250.6337

median(Co2data_Copy$Co2emmissions) #250.5847
median(Co2data_Copy$co2emission_model2) #250.6337


################################### 3rd Model ###########################################################
```

```r
################################### 3rd Model ###########################################################

lm_model3=lm(Co2emmissions~Enginesize,Co2data_Copy)
summary(lm_model3)

Coeff3=summary(lm_model3)$coefficients[1, 1]
Coeff3_Enginesize=summary(lm_model3)$coefficients[2, 1]
residuals_SD3=summary(lm_model3)$sigma #30.71721

######## test for normality of residuals #####
qqnorm(lm_model3$resid)
qqline(lm_model3$resid)

set.seed(100) #set seed
residualSD3_simulateddata=rnorm(n=nrow(Co2data_Copy),mean=0,sd=residuals_SD3) # simulated residuals using mean and sd for model 3
sum(residualSD3_simulateddata) #529.3767

##### populate the values co2 using model 3
for (i in c(1:nrow(Co2data_Copy)))
{

  Co2data_Copy[i,c('co2emission_model3')]=Coeff3+(
    Coeff3_Enginesize*Co2data_Copy[i,c('Enginesize')])+residualSD3_simulateddata[i] # iterating through linear model and residuals t
}


Co2data_Copy['co2emission_model3']=round(Co2data_Copy['co2emission_model3'])

lm_model3_sim=lm(co2emission_model3~Enginesize,Co2data_Copy)
summary(lm_model3_sim)
##### histogram for residuals of real and simulated data ########
par(mfrow=c(1,2))
hist(lm_model3$resid, main="Histogram of of Residuals for Model 3(EngineSize)",
     ylab="Frequency",xlab='Residuals',col="light pink")

hist(lm_model3_sim$resid, main="Histogram of simulated Residuals for Model 3 (EngineSize)",
     ylab="Frequency",xlab='Residuals',col="light pink") # newly simulated data
```