Phoenix Analytics

Mega Million Probabilities (2015-2016)

Group Members:
Inzamamdeen Kassim
Josue Criollo
Alexus Pargan
Christopher Triana
Estefany Gomez

Introduction:

The Mega Million consists of six numbers that a player can choose from or a machine can select at random. For our analysis, we assumed that all numbers were randomly selected without any user bias. This was a reasonable assumption since the number of players that select numbers using the machine is larger compared to the number of users that select their own numbers. Our analysis was conducted on the lottery drawings ranging from January, 2015 to December, 2016. During this time period, the first five numbers ranged from 1 to 75, and the sixth number, otherwise known as the Mega Ball, ranged from 1 to 15. Choosing the correct number for the Mega Ball automatically makes the player a winner, regardless of how many of the numbers were correct overall. However, they can also win by getting at least three numbers correct, assuming they did not guess the Mega Ball right. Overall, there are a total of nine ways to win, making the sample space for winning: W_{5+1},W_{5+0},W_{4+1},W_{4+0},W_{3+1}, W_{3+0},W_{2+1},W_{1+1},W_{0+1}, where the first digit is the amount of correct numbers from the pool of 75 possible numbers and the second digit is whether the Mega Ball is correct or not.

For each possible outcome in the sample space, there is a different prize that increases with the number of correct values. The prize can also vary due to the megaplier, which multiplies that prize by a factor of 2, 3, 4, or 5, if guessed correctly. Before applying the megaplier, the initial winning prizes are: Jackpot, $1,000,000, $5,000, $500, $50, $5 (for $W_{3+0}$and W_{2+1}). The cost for a regular Mega million ticket is $1 and for a Megaplier ticket is $2.

The probabilities of winning from 2015 to 2017 were determined using the hypergeometric distribution as followed:

$$P(W_{5+1}) = \frac{1}{N} = \frac{1}{258{,}890{,}850}$$

$$P(W_{5+0}) = \frac{m-1}{N} = \frac{1}{18{,}492{,}204}$$

$$P(W_{4+1}) = \frac{\binom{n-k}{1}\binom{k}{1}}{N} = \frac{1}{739{,}688}$$

$$P(W_{4+0}) = \frac{\binom{n-k}{1}\binom{k}{1}\binom{m-1}{1}}{N} = \frac{1}{52{,}835}$$

$$P(W_{3+1}) = \frac{\binom{n-k}{2}\binom{k}{2}}{N} = \frac{1}{10{,}720}$$

$$P(W_{3+0}) = \frac{\binom{n-k}{2}\binom{k}{2}\binom{m-1}{1}}{N} = \frac{1}{766}$$

$$P(W_{2+1}) = \frac{\binom{n-k}{3}\binom{k}{3}}{N} = \frac{1}{473}$$

$$P(W_{1+1}) = \frac{\binom{n-k}{4}\binom{k}{4}}{N} = \frac{1}{56}$$

$$P(W_{0+1}) = \frac{\binom{n-k}{5}\binom{k}{5}}{N} = \frac{1}{21}$$

$$P(\bar{W}_{2+0}) = \frac{\binom{n-k}{3}\binom{k}{3}\binom{m-1}{1}}{N} = \frac{1}{701.32}$$

$$P(\bar{W}_{1+0}) = \frac{\binom{n-k}{4}\binom{k}{4}\binom{m-1}{1}}{N} = \frac{1}{91.98}$$

$$P(\bar{W}_{0+0}) = \frac{\binom{n-k}{5}\binom{k}{5}\binom{m-1}{1}}{N} = \frac{1}{38.32}$$

Hypergeometric probabilities differ from binomial probabilities in that there is no replacement of a number as it is drawn. Since there is no replacement of a number once drawing the hypergeometric probability changes for the following number that will be drawn based on what was previously drawn. Unlike a binomial distribution where there is replacement, the probabilities remain constant for each drawing.

1.1 Empirical versus Theoretical Probabilities

In order for us to determine the empirical probabilities, we needed to determine the number of tickets sold for each drawings; since that information was not given we had to determine it from the theoretical probabilities listed above. This was done by finding the probability of winning any of the prizes (P(W)). Which was determined to be the sum of all the individual probabilities:

$P(W) = P(W_{5+1}) + P(W_{5+0})+P(W_{4+1})+P(W_{4+0})+P(W_{3+1})+P(W_{3+0})+P(W_{2+1})+P(W_{2+0})+P(W_{1+1})+P(W_{1+0})$

$$P(W) = \frac{1}{14.71}$$

Using this probability we were able to determine that there is approximately 1 winner for every 14.71 lottery tickets purchased. This allowed us to estimate the number of tickets sold by multiplying this ratio by the number of winners.

$$Number\ of\ \square ickets\ sold\ =\ Number\ of\ Winners\ *\ 14.71$$

Our next step was determining whether this approximation was valid by comparing the empirical and theoretical probabilities of winning a particular prize. R-Studio proved to be a useful tool in organizing these number of winners for a particular prize and calculating the probability for the respective prizes. This was done using the following formula:

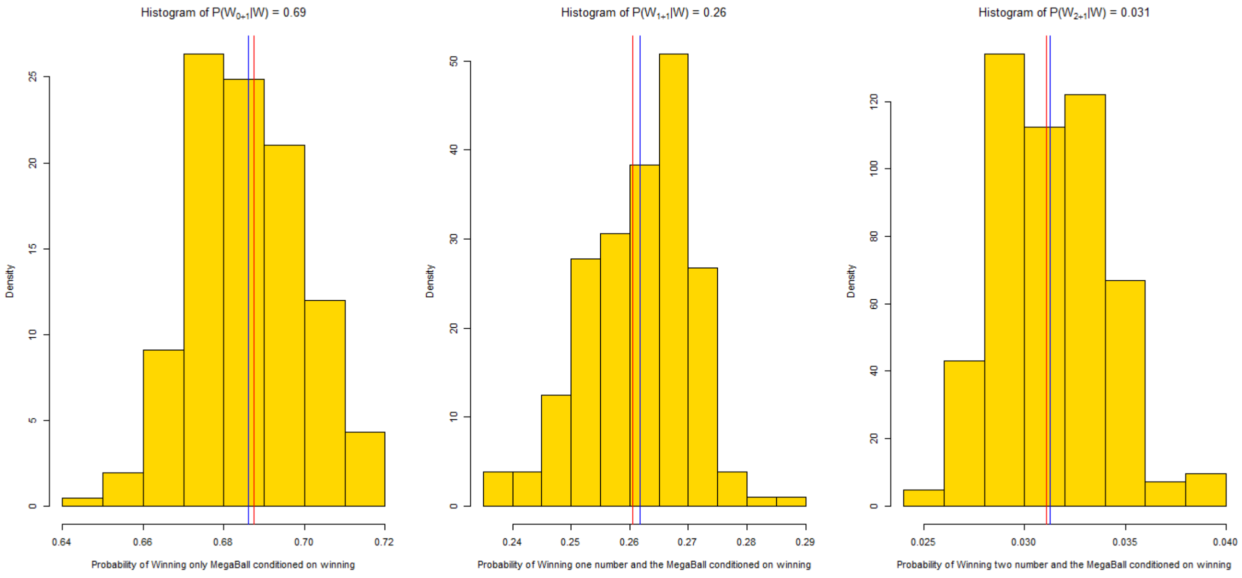
$$P(W_{0+1}|W) = \frac{P(W_{0+1}) \cap P(W)}{P(W)} = \frac{P(W_{0+1})}{P(W)}$$

This was done for each prize and each drawing, which we then calculated the Mean, the Standard Deviation, and the Coefficient of Variance. These values are listed below for $P(W_{2+1}|W)$, $P(W_{1+1}|W)$ and $P(W_{0+1}|W)$ since others were too small to notice any variations.

Probability	Mean ($\boldsymbol{\mu}$)	Standard Deviation (σ)	Coefficient of Variance (δ)
$P(W_{2+1}\|W)$	0.031	0.0026	0.085
$P(W_{1+1}\|W)$	0.26	0.0088	0.034
$P(W_{0+1}\|W)$	0.69	0.014	0.020

Table 1: Mean, standard deviation, and coefficient of variance of the actual probabilities $P(W_{2+1}|W)$, $P(W_{1+1}|W)$, and $P(W_{0+1}|W)$.

The Theoretical probabilities were calculated by comparing the respective probability formulas for each prize to the empirical probability using the following histograms. The y-axis shows the distribution of the probabilities for a particular prize being won for each drawing, whereas the x-axis shows the probability of winning that particular prize conditioned on winning. Each histogram shows several area regions and the probability of the player winning each particular prize is greater the higher the density. Using these histograms along with R-Studio, the group was able to calculate the mean, standard deviation, and coefficients of variation which would be the empirical probabilities whereas the theoretical probabilities would be referred to the hypergeometric distributions shown below. It can be seen that the theoretical probability is very close to the actual probability which confirms that our estimation for the number of tickets sold is valid.



(a.) (b.) (c.)

Graph 1. Histogram showing the distribution for (a.) $P(W_{0+1}|W)$=0.69, (b.) $P(W_{1+1}|W)$=0.26, and (c.) $P(W_{2+1}|W)$=0.031, where the x-axis is the probability of winning a specific criteria conditioned on winning and the y-axis is the density. Red vertical line represents the Theoretical probabilities, whiles the blue line represents the mean actual probabilities

1.2 Estimating Payout Amounts

The estimated payout was calculated by multiplying the normal winners by the prizes for each category and multiplying the Megaplier winners by the prizes won, then multiplying that by the megaplier number. Both of these payouts were added to the Jackpot payout, this provided us with the total payout per drawing.

Equations used:

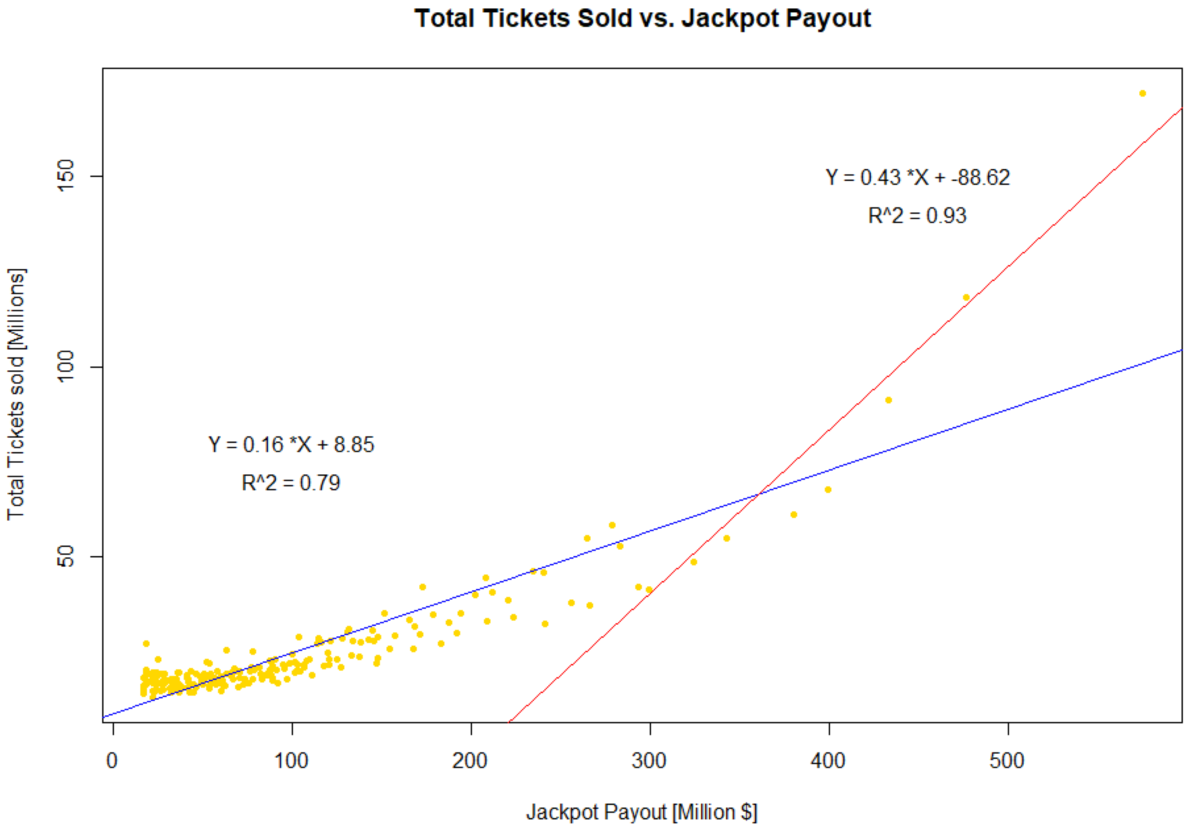
$$Normal\ Payout\ _{1+1} = Number\ of\ winner\ (W_{1+1}) \times Prize(W_{1+1})$$

$$Megaplier\ Payout\ = Number\ of\ winn\text{□}rs(W_{1+1}) \times Prize(W_{1+1}) \times Megaplier$$

$$Total\ Payout\ = Normal\ Payout + Megaplier\ Payout\ + Jackpot$$

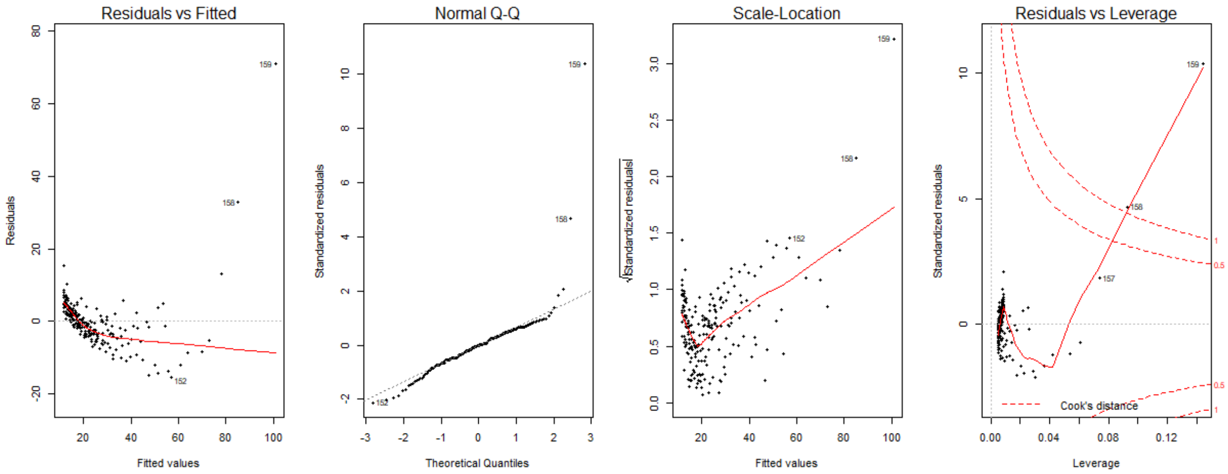
1.3 Number of Tickets Sold versus Jackpot Amount

The goal of this section is to analyze the relationship between the number of tickets sold and the Total Jackpot payout. Based on the linear regression plot in graph 2, it can be seen that as the jackpot amount increases, so does the number of tickets sold. From the scatter plot we could see that the plots follow two distinct linear trends, so we perform regression analysis below 280 Million jackpot payout and above 280 million.



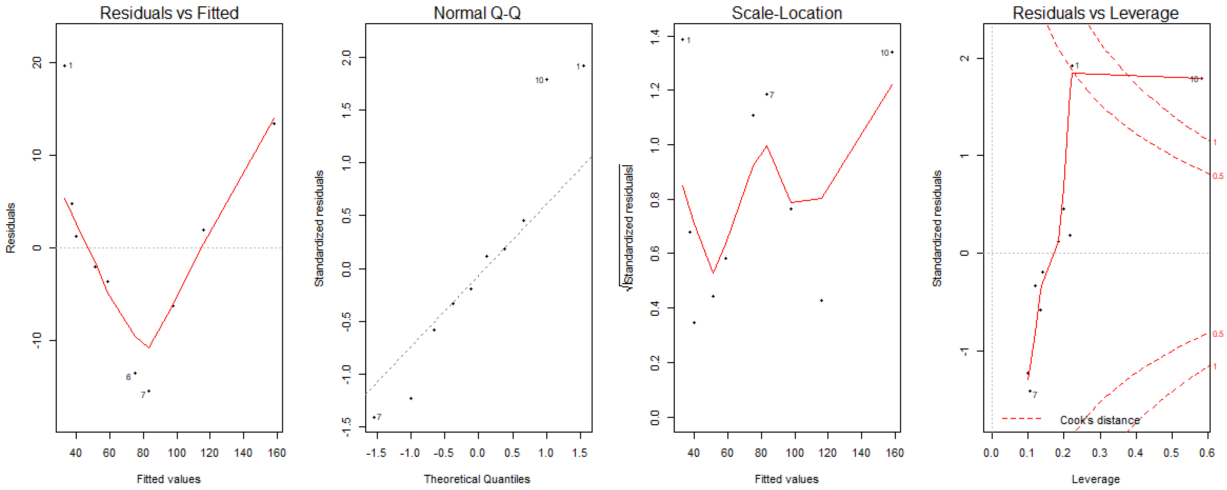
Graph 2: Scatter plot of Number of Ticket sold versus Jackpot Payout, where the x-axis is the Jackpot Payout (Millions USD) and the y-axis is the estimated Total Ticket sold (Millions).

Using R-Studio to create different graphs, we were able to analyze the relationship between the ticket sales and the jackpot amount using four methods: residual versus fitted, normal Q-Q, scale-location, and residual versus leverage. Two sets were made for when the jackpot was less than and greater than the $280 million threshold. In most of these graphs we were able to determine the linearity and variance of the plot, which is important since it determines the dispersity of the data, whether it is the mean or median. If there is a high variance, then there is a lot of differences in the data.



(a.) (b.) (c.) (d.)

Graph 3: Analysis of (a.) Residuals versus Fitted, (b.) Normal Q-Q, (c.) Scale-Location, and (d.) Residual versus Leverage relating the number of tickets sold to Jackpot less than $280.



(a.) (b.) (c.) (d.)

Graph 4: Analysis of (a.) Residuals versus Fitted, (b.) Normal Q-Q, (c.) Scale-Location, and (d.) Residual versus Leverage relating the number of tickets sold to Jackpot when it is more than $280.

Residuals versus Fitted

The first method used was residual versus fitted, as shown on graphs 3a and 4a. The purpose of these graphs was to determine any non-linearity from the residual, or errors, by “fitting”, or predicting a linear model. According to graph 3a, when the Jackpot is less than 280, it is clear that the number of sold tickets becomes less predictable as the jackpot increases. For example, when the fitted values are less than 20, they are clustered as a negative correlation, but become more scattered as it moves along the x-axis. Nonetheless, when the Jackpot is greater than 280, as seen in graph 4a, there are less values to draw any strong conclusions. However, they appear to be closer to a zero valued residual, compared to graph 3a, which may be due to the low value in data. If we analyze the graph 3a as a whole, it is noticeable that the range gradually increases in a blow horn shape. This means that the residuals, variance, errors, and even predictable values become larger, fitted values increase.

Normal Q-Q

Following residual versus fitted graphs is normal Q-Q, which plots standard residuals as a function of theoretical quantities. The Q-Q plot, or quantile-quantile plot, allows us to compare the residual to that of a normal distribution graph. The standard residual can be determined by dividing the standard deviation, determined in table 1, from the residuals. Both graphs 3b and 4b, display a positive linear slope, indicating that the relationship between ticket sales and Jackpot amount follows a normal distribution. Using normal Q-Q, we can verify the mean value, and extreme high and low, where the mean is located in the center and the extremes on the edges.

Scale-Location

Scale-Location can show the level of homoscedasticity, or equal variance, which indicates how far apart the numbers are spread from the mean. In addition, scale-location allows us to see if the residuals are evenly spread. Focusing on graphs 3c and 4c, we can see that both graphs are mainly increasing, which concludes that that there is a wide spread between residuals and there is an increase in variance. Overall, it confirms that the estimated ticket sales becomes less predictable as the Jackpot payout increases.

Residuals versus Leverage

Residuals versus Leverage allows us to determine influential points. Although, not all outliers (particular points that fall far from the other data points) are influential in the linear regression analysis. For example, in Graph 2, we see that the the point at the very top right corner has a high value of x and is also very far for the many measured points; the displayed high leverage suggests an influence over the regression of this graph. High leverage points have a greater ability to skew the line, and so the slope of the blue line is very different from the slope of the red line. The cause of this is due to the removal of this point which means the line wants to compensate for the loss of that particular point.

Not always will we find plots with a specific pattern. Based on our results, we found that when the number of tickets sold was less than $280 million, the “Residuals vs Leverage” graph was distinct from when the number of tickets sold was more than $280 million. Based on graphs 3d and 4d, we can say that they were out of proportion to the jackpot payout relative to the trend of the other data points, because they have high Cook`s distance scores; this means that the cases are influential to the regression results.

In graph 3a, we can identify influential cases because the point 149 is very far out of the red dashed line (Cook`s distance lines). From the scale used on the graph, the majority of the values are very close to each other and well within the boundaries of Cook`s distance. On the other hand, graph 4a has a case value that is right on Cook`s distance lines and another point that is outside of the cook`s boundaries.

Conclusion

In completing this evaluation, we were able to apply our fundamental statistical analysis skills in a manner that produced a realistic representation of the Mega Million lottery. From our exploration, the group learned that there are different tools that can be used to assess data on a large scale. Regression analysis is a tool used to make other conclusions about data beyond the scope of the initial set we were given. The various findings based on the data that was analyzed suggest that statistical examination of data allows us to develop conclusive and reliable understandings about the behavior of the data set as a whole. We were able to conclude that our plot which depicts the Number of Tickets Sold as a function of Jackpot Payout did not follow a linear relationship since our residual plot followed a pattern that was not randomly distributed. This was possibly due to another variable that we did not take into account when making our regression model.

Index (Code)

```
#This statement imports data from text file
megamillion = read.table("megamillionupdate.txt", header=T)
#Ratio of winners to population this will be used to estimate the number of tickets sold
Ratio=14.71
drawingwinners=matrix(NA,ncol=1,nrow=209)
#This loops calculates the sum of all the winners per drawing
for(i in 1:209){
 winnersum=0
  for(j in 4:20){
    winnersum=winnersum+megamillion[i,j]
  }
  drawingwinners[i,]=winnersum
}
colnames(drawingwinners)<- "Sum of Drawing Winners"
#The section below will calculate the number of tickets sold each drawing using the ratio of winners to population
numticketssold=matrix(NA,ncol=2, nrow=209)

for(i in 1:209){
  numticketssold[i,]=drawingwinners[i,]*Ratio
}
numticketssold[,2] <- numticketssold[,1]/1000000
colnames(numticketssold) <- c("Number of Tickets Sold per Drawing","Number of TIckets Sold [Millions]")
#Sum of winners per category
megamillionwinners=matrix(NA,ncol=9,nrow=209)
#This loop adds the winners for the W5+1 to the first coloum of the megamillionwinners table
  megamillionwinners[,1]=megamillion[,4]
#This loop calculates the sum of the normal megamillion and the magaplier and puts the results in the megamillion winners table
for(i in 1:209){
  for(j in 2:9){
    megamillionwinners[i,j]=megamillion[i,j+3]+megamillion[i,j+11]
  }
}
colnames(megamillionwinners)<-c("W5+1","W5+0","W4+1","W4+0","W3+1","W3+0","W2+1","W1+1","W0+1")
```

```
#Calculating the actual probabilities for each category
EachPrizeProbability = matrix(NA, nrow = 209,ncol= 9 )

for(i in 1:209){
  for(j in 1:9){
    EachPrizeProbability[i,j] <- megamillionwinners[i,j]/drawingwinners[i]
  }
}
colnames(EachPrizeProbability)<- c("W5+1","W5+0","W4+1","W4+0","W3+1","W3+0","W2+1","W1+1","W0+1")
#Calculating the mean and Standard Deviation
WinningStats=matrix(NA,nrow = 3, ncol = 3)
WinningStats[1,1] <-mean(EachPrizeProbability[,7])
WinningStats[2,1] <-mean(EachPrizeProbability[,8])
WinningStats[3,1] <-mean(EachPrizeProbability[,9])
WinningStats[1,2] <- sd(EachPrizeProbability[,7])
WinningStats[2,2] <- sd(EachPrizeProbability[,8])
WinningStats[3,2] <- sd(EachPrizeProbability[,9])
rownames(WinningStats)<- c("W2+1","W1+1","W0+1")
colnames(WinningStats) <- c("Mean","Standard Deviation","Coefficient of Variance")
#Calculating Coefficient of Variance
WinningStats[1,3] <- WinningStats[1,2]/WinningStats[1,1]
WinningStats[2,3] <- WinningStats[2,2]/WinningStats[2,1]
WinningStats[3,3] <- WinningStats[3,2]/WinningStats[3,1]
###################################
#Computing the Theoretical values
N=258890850
Combinations <-c(1,14,350,4900,24150,338100,547400,4584475,12103014)
SumCombinations = sum(Combinations)
TheriticalProbabilities = matrix(NA,ncol = 9, nrow = 5)
colnames(TheriticalProbabilities)<- c("W5+1","W5+0","W4+1","W4+0","W3+1","W3+0","W2+1","W1+1","W0+1")
for(i in 1:9){
  TheriticalProbabilities[1,i] <- (Combinations[i]/N)/(SumCombinations/N)
}
#Confidence 99%
Confidence = matrix(NA,ncol = 2, nrow = 3)
for(i in 7:9){
  Confidence[(i-6),1] <- TheriticalProbabilities[1,i]+(TheriticalProbabilities[1,i]*0.495)
  Confidence[(i-6),2] <- TheriticalProbabilities[1,i]-(TheriticalProbabilities[1,i]*0.495)
}
#Plotting Histogram function
par(mfrow=c(1,3))
hist(EachPrizeProbability[,9], probability = T, xlab = "Probability of Winning only MegaBall conditioned on winning", main = expression('Histogram of P(W'[0+1]*'|W) = 0.69'), col = "Gold")
abline(v=TheriticalProbabilities[1,9], col = "Red")
abline(v=WinningStats[3,1], col = "Blue")
hist(EachPrizeProbability[,8], probability = T, xlab = "Probability of Winning one number and the MegaBall conditioned on winning", main = expression('Histogram of P(W'[1+1]*'|W) = 0.26'), col = "Gold")
abline(v=TheriticalProbabilities[1,8], col = "Red")
abline(v=WinningStats[2,1], col = "Blue")
hist(EachPrizeProbability[,7], probability = T, xlab = "Probability of Winning two number and the MegaBall conditioned on winning", main = expression('Histogram of P(W'[2+1]*'|W) = 0.031'), col = "Gold")
abline(v=TheriticalProbabilities[1,7], col = "Red")
abline(v=WinningStats[1,1], col = "Blue")
########################################################################
#Jackpot payout
Payout = matrix(NA, nrow=209, ncol = 16)
prizes = c(1000000,5000,500,50,5,5,1,1)
########################################################################
#Payout for normal winners
for(i in 1:209){
  for(j in 1:8){
    Payout[i,j] <- (prizes[j]*megamillion[i,(j+4)])
  }
}
#####################################################################
#####################################################################
#Payout for Megaplier winners
```

```
##############################################################################
for(i in 1:209){
  for(j in 1:8){
    Payout[i,(j+8)] <- (prizes[j]*megamillion[i,(j+12)])*megamillion[i,2]

  }
}
colnames(Payout)<-c("W5+0","W4+1","W4+0","W3+1","W3+0","W2+1","W1+1","W0+1","M5+0","M4+1","M4+0","M3+1","M3+0","M2+1","M1+1","M0+1")
#sum of payout
PayoutSum = matrix(NA,nrow=209,ncol=3)
for (i in 1:209) {
  PayoutSum[i,1] = sum(Payout[i,])
}
PayoutSum[,2] = PayoutSum[,1]/1000000
PayoutSum[,3] = PayoutSum[,2]+ megamillion[,3]
colnames(PayoutSum)<- c("Payout Sum","Payout without Jackpot", "Payout Sum [Millions]")
##############################################################################
#Plot of NUmber of Tickets sold vs Jackpot
x = PayoutSum[,3]
y = numticketssold[,2]
par(mfrow=c(1,1))
plot(x,y, xlab = "Jackpot Payout [Million $]", ylab = "Total Tickets sold [Millions]", pch = 20, col = "Black", main = "Total Tickets Sold vs. Jackpot Payout")
Data = matrix(NA,nrow = 209, ncol = 2)
Data[,1] = x
Data[,2] = y
#lower regression
Data1 <- Data
Data1 <- Data1[c(Data1[1]<=280),]
Regression1 = lm(Data1[,2]~Data1[,1])
abline(Regression1, col = "Blue")
#upper Regression
Data2 <- Data
sumRegression1 = summary(Regression1)
```

```r
sumRegression2 = summary(Regression2)
Data2 <- Data2[c(Data2[,1]>280),]
Regression2 = lm(Data2[,2]~Data2[,1])
abline(Regression2, col = "Red")
text(100,80,paste("Y =", round(Regression1$coefficients[2], digits = 2),"*X +",round(Regression1$coefficients[1],digits = 2)))
text(100, 70,paste("R^2 =",round(sumRegression1$r.squared, digits = 2)))
text(450,150,paste("Y =", round(Regression2$coefficients[2], digits = 2),"*X +",round(Regression2$coefficients[1],digits = 2)))
text(450, 140,paste("R^2 =",round(sumRegression2$r.squared, digits = 2)))
##################################################################
#Regression Analysis
par(mfrow=c(2,4))
plot(Regression1, pch = 20, col = "Gold")
plot(Regression2, pch = 20, col = "Gold")
```