0.1 Put in Raw Data

```
Game <- c(1:23) #game number
Made <- c(4, 5, 5, 5, 2, 7, 6, 9,
          4, 1, 13, 5, 6, 9, 7, 3,
          8, 1, 18, 3, 10, 1, 3) #number of free throws made
Attempt <- c(5, 11, 14, 12, 7, 10, 14, 15,
             12, 4, 27, 17, 12, 9, 12, 10,
             12, 6, 39, 13, 17, 6 ,12) #number of free throw attempt
Miss <- Attempt - Made #number of free throws missed
```

0.2 Convert Data to Binary Data

```
l <- length(Game) #23
FT <- c() #free throws vector (binary: 0 or 1)
G <- c() #game vector
for (i in 1:l){
    made.i <- c(rep(1, Made[i]))
    miss.i <- c(rep(0, Miss[i]))
    new.game <- c(rep(Game[i], length(made.i)+length(miss.i)))
    FT <- c(FT, made.i, miss.i)
    G <- c(G, new.game)}
df.data <- data.frame("FT"=FT, "G"=G)
```

1 Constant Probability

1.1 Estimate α

```
reg <- glm(FT~1, family=binomial(link="identity"))
alpha <- summary(reg)$coef[1,1] #pull out interecept estimation
se.alpha <- summary(reg)$coef[1,2]#pull out standard error of incetercept estimate
CI.alpha <- confint(reg)
print.CI.alpha <- paste(decimal(CI.alpha, dec), collapse=",")
```

```
name <- c("$\\hat{\\alpha}$"
          , "Standard Error"
          , "Confidence Interval")
value <- c(decimal(alpha, dec),
           decimal(se.alpha, dec),
           paste("(", print.CI.alpha, ")"))
alpha.table <- data.frame("Statistic" = name, "Values" = value)
caption.alpha.table <- c("Constant Probability Model -  $\\alpha$")
knitr::kable(alpha.table, align='lr',caption=paste(caption.alpha.table))
```

Table 1: Constant Probability Model - α

Statistic	Values
$\hat{\alpha}$	0.456
Standard Error	0.029
Confidence Interval	(0.400,0.513)

1.2 Global Lack of Fit - Chi-Squared Approximation

```
n.star <- sum(Attempt)
dim.H0H1 <- length(Game)
paste("Check:", n.star, "=n.star >> dim(H0 U H1)=", dim.H0H1)
```

```
## [1] "Check: 296 =n.star >> dim(H0 U H1)= 23"
```

```
table <- cbind(Made, Miss)
chi.test <- chisq.test(table)
X2 <- chi.test$statistic
pval <- chi.test$p.value
df <- chi.test$parameter
paste("Pearson X-sq Statistic=", decimal(X2, dec), "on", df,
      "degrees of freedom, with p-value of", decimal(pval, dec))
```

```
## [1] "Pearson X-sq Statistic= 35.511 on 22 degrees of freedom, with p-value of 0.034"
```

1.3 Local Lack of Fit - Standardized Residuals

```
stdreschi <- chi.test$stdres[,1]#standardize residual
big.res <- length(stdreschi[abs(stdreschi)> 2])
res.table <- cbind(Game, stdreschi)
res.table <- res.table[order(-abs(stdreschi)),]
  #put residuals in order from larges abs value to smallest abs value
colnames(res.table) <- c("Game", "Standardized Residuals")
caption.res <- c("Constant Probability Model: Standardized Residuals")
res.table #sorted from largest to smallest abs value
```

```
##       Game Standardized Residuals
##  [1,]   14             3.32714873
##  [2,]   20            -1.66809291
##  [3,]    6             1.57550655
##  [4,]    1             1.55722841
##  [5,]   17             1.49526386
##  [6,]   23            -1.46327961
##  [7,]   18            -1.43798557
##  [8,]   22            -1.43798557
##  [9,]   12            -1.38101082
## [10,]    8             1.14859812
## [11,]   21             1.12683705
## [12,]   16            -1.00814962
## [13,]    5            -0.91588890
## [14,]   15             0.90355517
## [15,]    9            -0.87157092
## [16,]   10            -0.83317156
## [17,]    3            -0.76148494
## [18,]   13             0.31184647
## [19,]    4            -0.27986222
## [20,]   11             0.27797392
## [21,]    7            -0.21172996
## [22,]   19             0.07343578
## [23,]    2            -0.01042119
```

1.4 Game 14 - Outlier causing lack of fit?

```
table <- cbind(Made, Miss)
table.no14 <- table[-14,]
chi.test.no14 <- chisq.test(table.no14)
X2.no14 <- chi.test.no14$statistic
pval.no14 <- chi.test.no14$p.value
df.no14 <- chi.test.no14$parameter
paste("Pearson X-sq Statistic=" , decimal(X2.no14, dec), "on"
      , df.no14, "degrees of freedom, with p-value of", decimal(pval.no14, dec))
```

```
## [1] "Pearson X-sq Statistic= 24.618 on 21 degrees of freedom, with p-value of 0.264"
```

```
stdreschi <- chi.test$stdres[,1]#standardize residual
decimal(max(abs(stdreschi)), dec) #make standardize residual in absolute vaule
```

```
## [1] "3.327"
```

2 Constant Probability Model with Over-disperion

2.1 Estimate Dispersion and Scale

```
phi <- X2/ df
paste("X2 is", decimal(X2, dec), "on", df, "df --and so phi="
      , decimal(phi, dec), "and so sqrt(phi)=", decimal(sqrt(phi), dec))
```

```
## [1] "X2 is 35.511 on 22 df --and so phi= 1.614 and so sqrt(phi)= 1.270"
```

2.2 Estimate α, assuming $\phi \neq 1$

```
reg <- glm(FT~1, family=binomial(link="identity"))
reg.phi<-  summary(reg, dispersion=phi)
alpha.phi <- reg.phi$coef[1,1]
alpha.se.phi <- reg.phi$coef[1,2]

z.star <- qnorm(0.975)
CI.phi <- cbind(alpha.phi - z.star*alpha.se.phi, alpha + z.star*alpha.se.phi)
print.CI.phi <- paste(decimal(CI.phi, dec), collapse=",")
```

```
name <- c("$\\hat{\\alpha}$", "Standard Error of $\\alpha$", "Confidence Interval for $\\alpha$")
value <- c(decimal(alpha.phi, dec),  decimal(alpha.se.phi, dec),
           paste("(", print.CI.phi, ")"))
phi.table <- data.frame("Statistic" = name, "Values" = value)
caption.phi.table <- c("Constant Probability Model assuming $\\phi \\neq 1$ - $\\alpha$")
knitr::kable(phi.table, align="lr", caption=paste(caption.phi.table))
```

Table 2: Constant Probability Model assuming $\phi \neq 1$ - α

Statistic	Values
$\hat{\alpha}$	0.456
Standard Error of α	0.037
Confidence Interval for α	(0.384,0.528)