Class 6 Testing Graduations (Solution)

Yubo Rasmussen

## Introduction

This notebook will start with a section on standard table graduation, followed by testing graduation and goodness of fits.

## Standard Table Graduation Standard.r

This section illustrates standard table graduation using the **Gompertz model with Poisson errors** and compares different modeling approaches applied to **CMI data**. First, we load the required dataset, which contains mortality data for different years and age groups.

# URL for CMI\_read.r on github  
CMI\_Deaths\_url <- "https://raw.githubusercontent.com/yubae-bit/F79SU/main/CMI%20and%20HMD%20data%20sets/CMI\_Deaths.csv"  
CMI\_Exposures\_url <- "https://raw.githubusercontent.com/yubae-bit/F79SU/main/CMI%20and%20HMD%20data%20sets/CMI\_Exposures.csv"  
CMI\_Read\_url <- "https://raw.githubusercontent.com/yubae-bit/F79SU/main/CMI%20and%20HMD%20data%20sets/CMI\_read.r"  
  
# Download the necessary files  
download.file(CMI\_Deaths\_url, destfile = "CMI\_Deaths.csv", mode = "wb")  
download.file(CMI\_Exposures\_url, destfile = "CMI\_Exposures.csv", mode = "wb")  
  
# Now source the R script  
source("CMI\_read.r")

### Select Data for Standard Table Graduation

We begin by selecting mortality data for the year 2000, focusing on ages 40 to 80. This data will be used to fit our standard table graduation model.

DTH.s <- Dth[ (40 <= Age) & (Age <= 80), Year == 2000]  
EXP.s <- Exp[ (40 <= Age) & (Age <= 80), Year == 2000]  
Obs.s <- log(DTH.s/EXP.s)  
AGE <- 40:80

### Fit Gompertz Model with Poisson Errors

Now, we fit a Poisson regression model to estimate mortality rates using the Gompertz function. The offset for exposure ensures proper modeling of mortality rates per unit exposure.

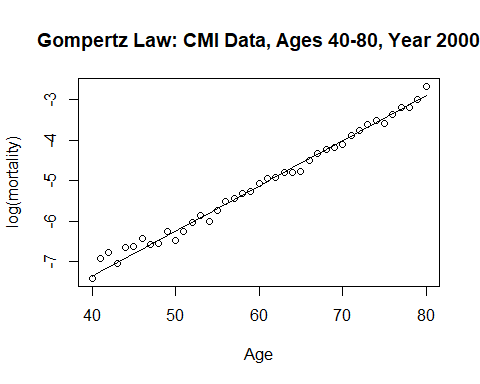
Gomp.s <- glm(DTH.s ~ offset(log(EXP.s)) + AGE, family = poisson)  
summary(Gomp.s)

Call:  
glm(formula = DTH.s ~ offset(log(EXP.s)) + AGE, family = poisson)  
  
Deviance Residuals:   
 Min 1Q Median 3Q Max   
-3.2944 -0.5540 0.0899 1.0603 4.0042   
  
Coefficients:  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) -11.823753 0.072473 -163.15 <2e-16 \*\*\*  
AGE 0.111544 0.001128 98.91 <2e-16 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
(Dispersion parameter for poisson family taken to be 1)  
  
 Null deviance: 9482.758 on 40 degrees of freedom  
Residual deviance: 88.447 on 39 degrees of freedom  
AIC: 379.76  
  
Number of Fisher Scoring iterations: 4

### Visualization of Standard Table Graduation

The fitted model is plotted to visualize the estimated mortality rates against observed values. This helps assess how well the model fits the data.

plot(AGE, Obs.s, xlab = "Age", ylab = "log(mortality)",  
 main = "Gompertz Law: CMI Data, Ages 40-80, Year 2000")  
lines(AGE, Gomp.s$lin - log(EXP.s))



### Save Standard Table Graduation on Log Scale

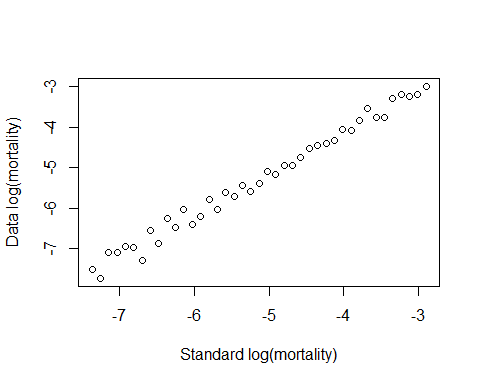
We store the fitted log mortality rates for further comparison with other datasets.

log.mu.s <- Gomp.s$lin - log(EXP.s)

### Comparison with Fresh Data (Year 2005)

Next, we retrieve mortality data for the year 2005 and compare it against the standard table from 2000 to observe potential mortality trends.

DTH <- Dth[ (40 <= Age) & (Age <= 80), Year == 2005]  
EXP <- Exp[ (40 <= Age) & (Age <= 80), Year == 2005]  
Obs <- log(DTH/EXP)  
plot(log.mu.s, Obs, xlab = "Standard log(mortality)",  
 ylab = "Data log(mortality)")



### Graduation of 2005 Data using Gompertz Model

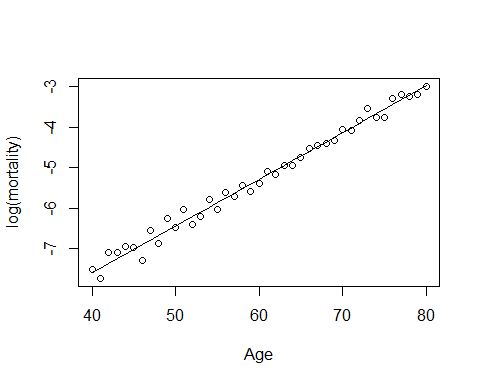
To understand how mortality rates evolve, we fit a new Gompertz model for the 2005 dataset and compare it with the standard table graduation.

Gomp.2005 <- glm(DTH ~ offset(log(EXP)) + AGE, family = poisson)

### Fit GLM of DTH on Standard Table

Now, we fit a generalized linear model (GLM) to estimate the relationship between deaths and the previously fitted standard table log mortality rates.

Grad <- glm(DTH ~ log.mu.s + offset(log(EXP)), family = poisson)  
plot(AGE, Obs, xlab = "Age", ylab = "log(mortality)")  
lines(AGE, Grad$lin - log(EXP))



### Weighted Least Squares Graduation

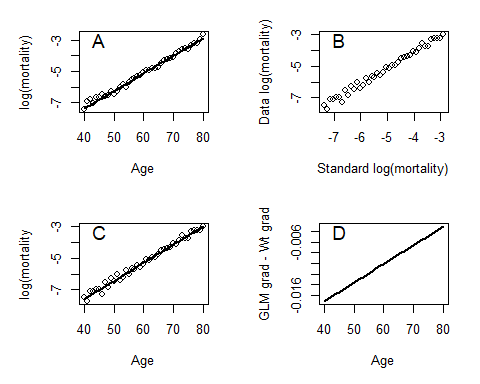
In addition to the Poisson regression approach, we perform a weighted least squares regression to account for variability in exposure.

Grad.wt <- lm(Obs ~ log.mu.s, weights = DTH)

### Comparison of Standard and Data Graduation

We compare different graduation approaches through multiple visualizations to assess the effectiveness of different modeling techniques.

par(mfrow = c(2,2))  
par(mar = c(5, 5, 2, 2))  
  
plot(AGE, Obs.s, xlab = "Age", ylab = "log(mortality)")  
lines(AGE, log.mu.s, lwd = 2)  
text(45, -3, "A", cex = 1.4)  
  
plot(log.mu.s, Obs, xlab = "Standard log(mortality)",  
 ylab = "Data log(mortality)")  
text(-6.8, -3.3, "B", cex = 1.4)  
  
plot(AGE, Obs, xlab = "Age", ylab = "log(mortality")  
lines(AGE, Grad$lin - log(EXP), lwd = 2)  
text(45, -3.3, "C", cex = 1.4)  
  
plot(AGE, Grad$lin - log(EXP) - Grad.wt$fit, xlab = "Age",  
 ylab = "GLM grad - Wt grad", type = "l", lwd = 2)  
text(45, -0.004, "D", cex = 1.4)



#### Interpretation of the plots

* **Panels A & C** show that both **Gompertz and GLM models fit the mortality data well**.
* **Panel B** confirms that **the standard table graduation closely follows observed data**.
* **Panel D** highlights **differences between GLM and weighted least squares approaches**, which may suggest potential refinements in modeling mortality trends.

### Conclusion

* The Gompertz Poisson model is effective for mortality graduation.
* Comparison of 2000 and 2005 data allows assessment of changes in mortality trends.
* Weighted least squares provides an alternative fit that accounts for exposure size.

## Functions for Testing Graduation Test\_GoF.r

### Chi-squared

# Chi-square test  
#  
# Arguments: Z = residuals Z.x  
# Npar = number of fitted parameters  
#  
Chi.Square <- function(Z, Npar){  
 Chis2 <- sum(Z^2)  
 DF <- length(Z) - Npar  
 Sig.Pr <- 1 - pchisq(Chis2, DF)  
 return(list(Chis2 = Chis2, DF = DF, Sig.Pr = Sig.Pr))  
}

### Standardized deviations test

#### Equal Area Test

# Equal area test  
#  
# Arguments: Z = residuals Z.x  
# N = Number of cells  
#  
Standard.Area <- function(Z, N){  
 DF <- N-1  
 Exp <- length(Z)/N  
 CDF <- seq(0, 1, length = (N+1))  
 Inner.Boundary <- qnorm(CDF[-c(1, length(CDF))])  
 Boundary <- c(-20, Inner.Boundary, 20)  
 Obs <- hist(Z, breaks = Boundary, plot = FALSE)$counts  
 Chis2 <- sum( (Obs - Exp)^2/Exp)  
 Sig.Pr <- 1 - pchisq(Chis2, DF)  
 return(list(Boundary = round(Inner.Boundary, digits = 3), Obs = Obs, Exp = Exp,  
 DF = DF, Chis2 = Chis2, Sig.Pr = Sig.Pr))  
}

#### Equal Width Test

# Equal width test  
#  
# Arguments: Z = residuals Z.x  
# W = Width of internal cells  
#  
Standard.Width <- function(Z, W){  
#  
# Calculate position of upper boundary and cell boundaries  
#  
 n <- length(Z)  
 Upper <- qnorm(1 - 5/n)  
 Upper <- max(seq(0, Upper, by = W))  
 Inner.Boundary <- seq(-Upper, Upper, by = W)  
 Boundary <- c(-20, Inner.Boundary, 20)  
#  
# Calculate DF and O\_i  
#  
 DF <- length(Boundary) - 2  
 Obs <- hist(Z, breaks = Boundary, plot = FALSE)$counts  
#  
# Calculate E\_i  
#  
 CDF.p <- c(0, pnorm(Inner.Boundary), 1) # Cumulative probabilities  
 CDF.f <- n \* CDF.p # Cumulative frequencies  
 Exp <- diff(CDF.f) # E\_i  
#  
# Compute Chi^2 & sig prob  
#  
 Chis2 <- sum( (Obs - Exp)^2/Exp)  
 Sig.Pr <- 1 - pchisq(Chis2, DF)  
 return(list(Boundary = round(Inner.Boundary, digits = 3), Obs = Obs, Exp = Exp,  
 DF = DF, Chis2 = Chis2, Sig.Pr = Sig.Pr))  
}

### Sign Test

# Sign test  
#  
# Argument: Z = residuals Z.x  
#  
Sign <- function(Z){  
 n <- length(Z)  
 Greater <- sum(Z >= 0)  
 Less <- sum(Z < 0)  
 if(Greater == Less){  
 return(list(N.plus = Greater, N.minus = Less, Sig.Prob = 1))  
 }  
 if(Greater > Less){  
 Sig.Prob <- 2 \* (1 - pbinom(Greater - 1, n, 0.5))  
 return(list(N.plus = Greater, N.minus = Less, Sig.Prob = Sig.Prob))  
 }  
 if(Greater < Less){  
 Sig.Prob <- 2 \* pbinom(Greater, n, 0.5)  
 return(list(N.plus = Greater, N.minus = Less, Sig.Prob = Sig.Prob))  
 }  
}

### Change of Sign Test

# Change of sign test  
#  
# Argument: Z = residuals Z.x  
#  
Change.Sign <- function(Z){  
 Change <- 0  
 N1 <- length(Z) - 1  
 for(i in 1:N1) {  
 if(Z[i] \* Z[i+1] < 0) Change <- Change + 1  
 }  
 Sig.Pr <- pbinom(Change, N1, 0.5)  
 return(list(N = N1+1, Change = Change, Sig.Pr = Sig.Pr))  
}

### Runs Test

# Runs test  
#  
# Arguments: Z = residuals Z.x  
#  
# This function calls the subsiduary function Runs( )  
#  
Runs.test <- function(Z){  
 Code <- Z; Code[ Z <= 0] = -1; Code[ Z > 0] = 1  
 n1 <- sum(Code == 1); n2 <- sum(Code == -1)  
 g <- 0; if(Code[1] > 0) g <- 1  
 for(i in 1:(length(Code)-1)) if(Code[i] < Code[i+1]) g <- g + 1  
 Sig.Pr <- Runs(n1, n2, g)  
 return(list(n1 = n1, n2 = n2, g = g, Sig.Prob = Sig.Pr))  
}  
#  
# Subsiduary function to calculate significance probability  
#  
Runs <- function(n1, n2, g){  
 Sig.Pr <- 0  
 Denom <- choose(n1+n2, n1)  
 for(i in 1:g){  
 Num <- choose(n1-1,i-1)\*choose(n2+1,i)  
 Sig.Pr <- Sig.Pr + Num/Denom  
 }  
 Sig.Pr  
}

#### Using a Permutation Test

# Runs test using a permutation test  
#  
# Arguments: Z = residuals Z.x  
# n = number of samples - defaults to 1000 if n is not set  
# Calls the function Runs.test( )  
#  
Runs.Test.Perm <- function(Z, n = 1000){  
 Code <- Z; Code[ Z <= 0] = -1; Code[ Z > 0] = 1  
 g <- Runs.test(Z)$g  
 Null.dist <- NULL  
 for(i in 1:n) {  
 Perm <- sample(Code)  
 Runs <- 0; if(Perm[1] > 0) Runs <- 1  
 for(i in 1:(length(Perm)-1)) if(Perm[i] < Perm[i+1]) Runs <- Runs + 1  
 Null.dist <- c(Null.dist, Runs)  
 }  
 Sig.Pr <- sum(Null.dist <= g)/n  
# Plot null distribution with observed g indicated  
 Null.dist <- table(Null.dist)/n  
 x <- as.numeric(names(Null.dist))  
 par(mfrow = c(1,1))  
 plot(x, Null.dist, type = "h", lwd = 2, col = "blue",  
 main = "Null distribution of runs statistics",  
 xlab = "Number of runs", ylab = "Probability")  
 lines(g, Null.dist[paste(g)], type = "h", col = "red", lwd = 3)  
 return(list(Runs = g, Null.dist = Null.dist, Sig.Pr = Sig.Pr))  
}

### Serial Correlation test

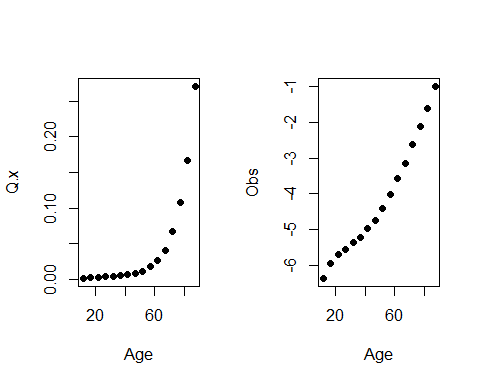
# Serial correlation test  
#  
# Arguments: Z = residuals Z.x  
#  
Serial <- function(Z){  
 Set1 <- Z[-length(Z)]; Set2 <- Z[-1]  
 Corr <- cor(Set1, Set2)  
 Sig.Pr <- 1 - pnorm(Corr, 0, sqrt(1/(length(Z)-1)))  
 return(list(Serial = Corr, Sig.Pr = Sig.Pr))  
}

## Example from Chapter 6 Chap\_6\_Eg.r

#  
# Example in Chapter 6 to illustrate statistical tests  
#  
# Input data, draw plots, fit model and calculate Z.x  
#  
Age <- seq(12,87, by = 5)  
E.x <- c(8119,7750,6525,5998,5586, 5245,4659,4222,3660,3012,  
2500,2113,1469,883,418, 181)  
D.x <- c(14,20,22,23,26, 28,32,37,44,54, 68,87,100,95,70,49)  
Q.x <- D.x/E.x  
cbind(Age, E.x, D.x, round(Q.x, digits = 4))

Age E.x D.x   
 [1,] 12 8119 14 0.0017  
 [2,] 17 7750 20 0.0026  
 [3,] 22 6525 22 0.0034  
 [4,] 27 5998 23 0.0038  
 [5,] 32 5586 26 0.0047  
 [6,] 37 5245 28 0.0053  
 [7,] 42 4659 32 0.0069  
 [8,] 47 4222 37 0.0088  
 [9,] 52 3660 44 0.0120  
[10,] 57 3012 54 0.0179  
[11,] 62 2500 68 0.0272  
[12,] 67 2113 87 0.0412  
[13,] 72 1469 100 0.0681  
[14,] 77 883 95 0.1076  
[15,] 82 418 70 0.1675  
[16,] 87 181 49 0.2707

Logit <- function(x) log(x/(1-x)) # Define logit function  
Obs <- Logit(Q.x)  
par(mfrow = c(1,2))  
plot(Age, Q.x, pch = 16)  
plot(Age, Obs, pch = 16)



Age2 <- Age^2  
Gomp.bin <- glm(Q.x ~ Age + Age2, weights = E.x, family = binomial)  
Q.dot <- Gomp.bin$fit  
Dth.dot <- E.x \* Q.dot  
Z.x <- (D.x - Dth.dot)/sqrt(Dth.dot)  
cbind(Age, E.x, D.x, round(Dth.dot, dig = 2), round(Z.x, dig = 3))

Age E.x D.x   
1 12 8119 14 18.78 -1.104  
2 17 7750 20 19.68 0.072  
3 22 6525 22 18.83 0.730  
4 27 5998 23 20.37 0.583  
5 32 5586 26 23.10 0.603  
6 37 5245 28 27.35 0.125  
7 42 4659 32 31.70 0.054  
8 47 4222 37 38.77 -0.285  
9 52 3660 44 46.92 -0.427  
10 57 3012 54 55.70 -0.228  
11 62 2500 68 68.81 -0.097  
12 67 2113 87 89.10 -0.222  
13 72 1469 100 97.26 0.278  
14 77 883 95 93.36 0.169  
15 82 418 70 70.87 -0.103  
16 87 181 49 48.40 0.086

#  
# Chis^2 test  
#  
Chi.Square(Z.x, 3)

$Chis2  
[1] 2.975883  
  
$DF  
[1] 13  
  
$Sig.Pr  
[1] 0.9980199

#  
# Standardised deviations test  
#  
# Equal width cells  
#  
Standard.Width(Z.x, 1)

$Boundary  
[1] 0  
  
$Obs  
[1] 7 9  
  
$Exp  
[1] 8 8  
  
$DF  
[1] 1  
  
$Chis2  
[1] 0.25  
  
$Sig.Pr  
[1] 0.6170751

#  
# Equal area cells  
#  
Standard.Area(Z.x, 3)

$Boundary  
[1] -0.431 0.431  
  
$Obs  
[1] 1 12 3  
  
$Exp  
[1] 5.333333  
  
$DF  
[1] 2  
  
$Chis2  
[1] 12.875  
  
$Sig.Pr  
[1] 0.001600403

#  
# Sign test  
#  
Sign(Z.x)

$N.plus  
[1] 9  
  
$N.minus  
[1] 7  
  
$Sig.Prob  
[1] 0.8036194

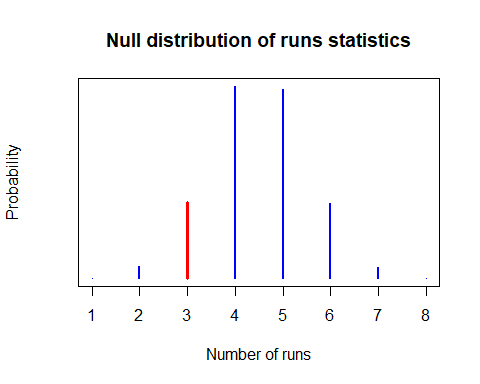
#  
# Change of sign test  
#  
Change.Sign(Z.x)

$N  
[1] 16  
  
$Change  
[1] 5  
  
$Sig.Pr  
[1] 0.1508789

#  
# Runs test  
#  
Runs.test(Z.x)

$n1  
[1] 9  
  
$n2  
[1] 7  
  
$g  
[1] 3  
  
$Sig.Prob  
[1] 0.1573427

#  
# and in its permutation form  
#  
Runs.Test.Perm(Z.x, 5000)



$Runs  
[1] 3  
  
$Null.dist  
Null.dist  
 1 2 3 4 5 6 7 8   
0.0006 0.0212 0.1372 0.3450 0.3394 0.1346 0.0210 0.0010   
  
$Sig.Pr  
[1] 0.159

#  
# Serial correlation test  
#  
Serial(Z.x)

$Serial  
[1] 0.4648827  
  
$Sig.Pr  
[1] 0.03589221

#

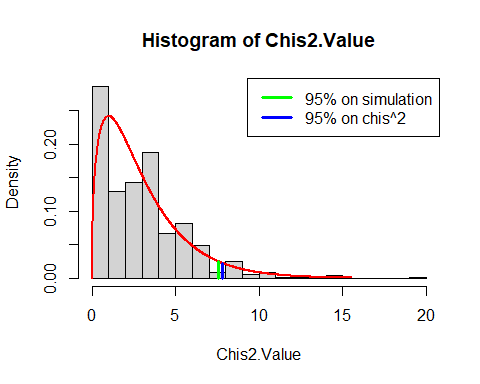
How would you interpret each test?

## Aside: Using simulations in ChiTest\_Sim.r

#  
# Chis2 tests or standardized deviations test via simulation  
#  
# Z contains the residuals  
# N is the number of cells  
#  
Chi2.Area <- function(Z, N){  
 DF <- N-1  
 Exp <- length(Z)/N  
 CDF <- seq(0, 1, length = (N+1))  
 Inner.Boundary <- qnorm(CDF[-c(1, length(CDF))])  
 Boundary <- c(-20, Inner.Boundary, 20)  
 Obs <- hist(Z, breaks = Boundary, plot = FALSE)$counts  
 sum( (Obs - Exp)^2/Exp)  
}  
#  
N.Sim <- 10000 # Number of simulations  
N.Z <- 11 # Number of residuals, Z.x  
N.Cells <- 4 # Number cells for test  
Chis2.Value <- numeric(N.Sim)  
for (i in 1:N.Sim) Chis2.Value[i] <- Chi2.Area(rnorm(N.Z), N.Cells)  
#  
Upper <- quantile(Chis2.Value, 0.95)  
c(Upper, pchisq(Upper, N.Cells -1) )

95% 95%   
7.545455 0.943598

#  
# We can compare the simulated null distribution with the  
# asymptotic chi^2 distribution  
#  
hist(Chis2.Value, breaks = 20, prob = T)   
x <- seq(0, c(quantile(Chis2.Value, 0.999)), len = 1000)  
lines(x, dchisq(x, N.Cells - 1), col = "red", lwd = 2)  
x1 <- c(quantile(Chis2.Value, 0.95))  
lines(c(x1,x1), c(0, dchisq(x1, N.Cells - 1)), lwd = 3, col = "green")  
x2 <- qchisq(0.95, N.Cells - 1)  
lines(c(x2,x2), c(0, dchisq(x2, N.Cells - 1)), lwd = 3, col = "blue")  
legend("topright", legend = c("95% on simulation", "95% on chis^2"),  
 col = c("green", "blue"), lwd = 3)



How would you interpret this?