Class 5 GLMs and Graduation

Yubo Rasmussen

## Introduction

This document explores Gompertz’s law of mortality using different modeling approaches, including normal models, Poisson regression, and binomial models (GLMs), applied to CMI data.

Load the data from my github repo

EngWales.url <- "https://raw.githubusercontent.com/yubae-bit/F79SU/main/CMI%20and%20HMD%20data%20sets/EngWales.Rdata"  
CMI.url <- "https://raw.githubusercontent.com/yubae-bit/F79SU/main/CMI%20and%20HMD%20data%20sets/CMI.Rdata"  
  
  
load(url(EngWales.url))  
load(url(CMI.url))

## Normal Models

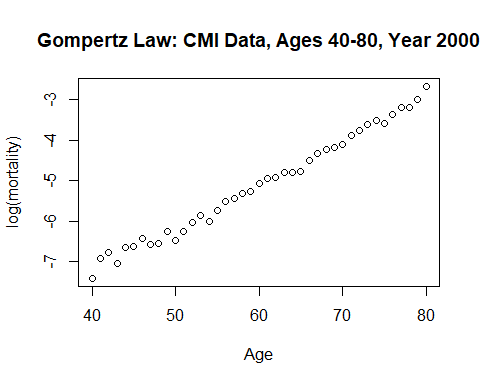
We begin by fitting a Normal model, i.e. a linear regression model. Can you recall what it looks like mathematically?

### Select Data for Year 2000, Ages 40-80

Your Task first is to select and store the number of deaths and exposure for ages 40-80 in 2000. Define a new variable “Obs” that adjusts for exposure and plot “Age” against “Obs”.

Q: The Gompertz law of mortality states that mortality rates increase exponentially with ago. Why do we use Obs <- log(DTH/EXP)?

DTH <- Dth[ (40 <= Age) & (Age <= 80), Year == 2000]  
EXP <- Exp[ (40 <= Age) & (Age <= 80), Year == 2000]  
AGE <- 40:80  
Obs <- log(DTH/EXP)  
plot(AGE, Obs, xlab = "Age", ylab = "log(mortality)",  
 main = "Gompertz Law: CMI Data, Ages 40-80, Year 2000")



### Fit Simple Normal Model

Let us proceed to fit a simple normal model. Using lm, proceed to fit Obs ~ AGE. Add the line of best fit from your regression model to the plot above. What are the coefficients of the fited model?

How would you interpret this model and what is ?

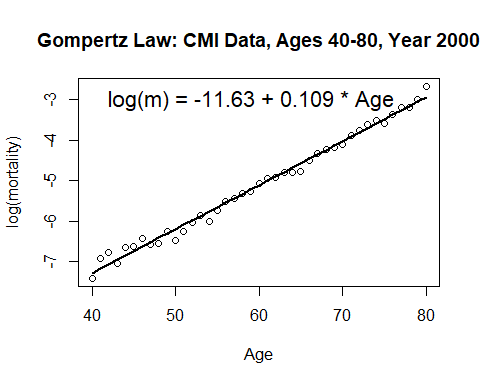
Gomp.fit <- lm(Obs ~ AGE)  
names(Gomp.fit)

[1] "coefficients" "residuals" "effects" "rank"   
 [5] "fitted.values" "assign" "qr" "df.residual"   
 [9] "xlevels" "call" "terms" "model"

plot(AGE, Obs, xlab = "Age", ylab = "log(mortality)",  
 main = "Gompertz Law: CMI Data, Ages 40-80, Year 2000")  
lines(AGE, Gomp.fit$fit, lwd = 2)  
Gomp.fit$coef

(Intercept) AGE   
-11.6285913 0.1086161

text(42, -3, "log(m) = -11.63 + 0.109 \* Age", adj = 0, cex = 1.4)

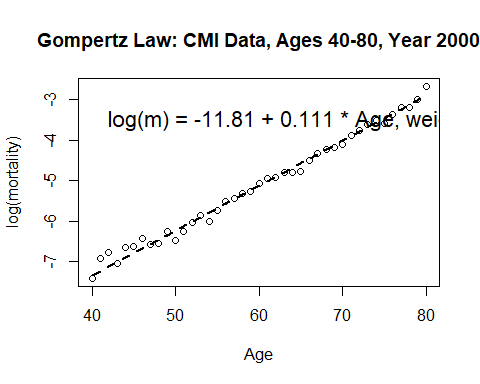


### Fit Weighted Normal Model

A simple normal model assumes that all observations are equally important, whereas a weighted normal model says otherwise. The weighted model assigns weights to observations based on the number of deaths, where ages with more deaths have a higher effect on the regression line, and vice versa.

Now you will proceed to fit a weighted normal model, and plot it as before. Are there any difference in the results between the two models?

Gomp.fit.w <- lm(Obs ~ AGE, weights = DTH)  
  
plot(AGE, Obs, xlab = "Age", ylab = "log(mortality)",  
 main = "Gompertz Law: CMI Data, Ages 40-80, Year 2000")  
lines(AGE, Gomp.fit.w$fit, lwd = 2, lty = 2)  
text(42, -3.5, "log(m) = -11.81 + 0.111 \* Age, weighted", adj = 0, cex = 1.4)

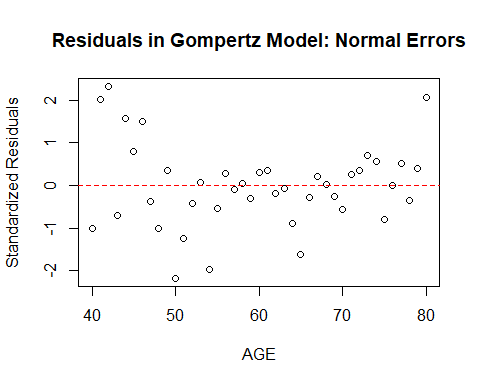


### Residual Analysis

Residuals represent the difference between the actual observed values and the predicted values from our model. Examining residuals helps us understand the goodness of fit and whether the assumptions of the model hold.

1. **Compute the standard deviation of residuals**: This is used to standardize the residuals.
2. **Compute the standardized residuals**: Defined as , where is the residual standard deviation.
3. **Plot residuals**: Helps identify potential outliers and check if residuals are randomly distributed (which supports model validity).

Sigma <- sqrt(sum(Gomp.fit$res^2)/Gomp.fit$df.res)  
Z.vals <- (Obs - Gomp.fit$fit)/Sigma  
plot(AGE, Z.vals, ylab = "Standardized Residuals",  
 main = "Residuals in Gompertz Model: Normal Errors")  
abline(h = 0, lty = 2, col = "red")



**What are your Interpretations of the residuals?**

* If the residuals are randomly scattered around zero, the model’s assumptions are likely valid, it doesn’t look like the normal assumption is valid here.
* Systematic patterns in the residual plot suggest potential model misspecifications, it seems like there is a quadratic pattern.
* Large residuals may indicate outliers or data issues, which is not an issue here.

## Poisson Model

In this section, we model mortality rates using a **Poisson Generalized Linear Model (GLM)** with a **log-link function**. The Poisson model assumes that the number of deaths follows a Poisson distribution:

where the **logarithm of the expected deaths** is modeled as a **linear function of age**:

Here, is the exposure (i.e., number of life-years at risk), included as an **offset** in the model.

### Fit the Poisson GLM Model

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

Call:  
glm(formula = DTH ~ offset(log(EXP)) + 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

* The offset(log(EXP)) ensures that the model adjusts for exposure.
* The estimated coefficients and determine the mortality curve.

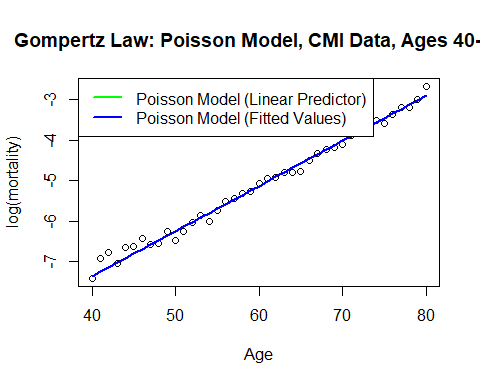
### Interpret the Coefficients

* The **intercept (**) represents the log mortality rate at the baseline age.
* The **slope (**) determines how mortality increases with age.
* A positive slope indicates that mortality increases exponentially over time, consistent with **Gompertz’s Law**.

### Plot the results for visualization

Visualize the **observed mortality rates** and compare them with **model predictions**.

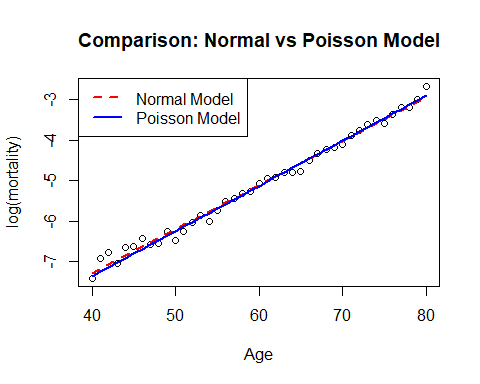
plot(AGE, Obs, xlab = "Age", ylab = "log(mortality)",  
 main = "Gompertz Law: Poisson Model, CMI Data, Ages 40-80")  
  
# Add the fitted Poisson regression line  
lines(AGE, Gomp.glm$coef[1] + Gomp.glm$coef[2]\*AGE, col = "green", lwd = 2)  
lines(AGE, log(Gomp.glm$fit/EXP), col = "blue", lwd = 2)  
  
# Add a legend for clarity  
legend("topleft", legend = c("Poisson Model (Linear Predictor)",   
 "Poisson Model (Fitted Values)"),   
 col = c("green", "blue"), lty = c(1, 1), lwd = 2)



#### Alternative Visualization: Comparing Poisson and Normal Model

Since we already fitted a **normal model (OLS regression)**, it is useful to **compare** it with the Poisson model.

plot(AGE, Obs, xlab = "Age", ylab = "log(mortality)",  
 main = "Comparison: Normal vs Poisson Model")  
  
lines(AGE, Gomp.fit$fit, col = "red", lwd = 2, lty = 2) # Normal Model  
lines(AGE, log(Gomp.glm$fit/EXP), col = "blue", lwd = 2) # Poisson Model  
  
legend("topleft", legend = c("Normal Model", "Poisson Model"),  
 col = c("red", "blue"), lty = c(2,1), lwd = 2)



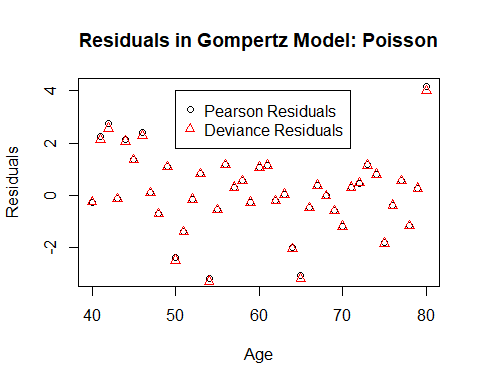
### Residual Analysis for Poisson Model

Residuals help us **assess model fit** and detect possible **misspecifications**.

#### Plot the Pearson and Deviance Residuals

* **Pearson residuals** measure how far observed values deviate from expected values, standardized by variance.
* **Deviance residuals** quantify goodness of fit by comparing observed and fitted values.

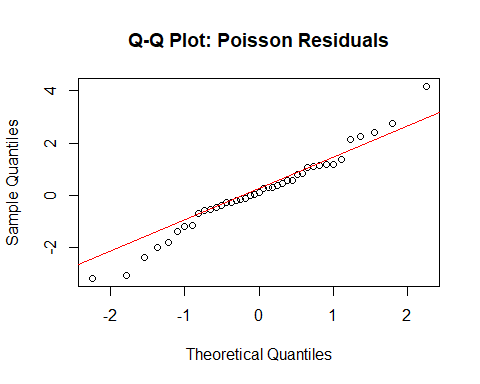
plot(AGE, resid(Gomp.glm, type = "pear"), xlab = "Age",  
 ylab = "Residuals", main = "Residuals in Gompertz Model: Poisson")  
points(AGE, resid(Gomp.glm, type = "deviance"), pch = 2, col = "red")  
legend(50, 4, legend = c("Pearson Residuals", "Deviance Residuals"),   
 pch = c(1, 2), col = c("black", "red"))



#### Alternative Residual Analysis: Q-Q Plot

A **Q-Q plot** compares the residuals to a normal distribution to check for **deviation from normality**.

qqnorm(resid(Gomp.glm, type = "pear"), main = "Q-Q Plot: Poisson Residuals")  
qqline(resid(Gomp.glm, type = "pear"), col = "red")



#### What are potential model refinements?

If residuals indicate poor fit:

1. **Quadratic Model:** Add an term to allow for curvature.

2. **Binomial Model:** If overdispersion is present, a **binomial** GLM may be more appropriate.

3. **Spline Regression:** A **generalized additive model (GAM)** could capture nonlinear trends.

## Binomial Model

Now, we move to the **Binomial Generalized Linear Model (GLM)** using a **logit-link function**. This model assumes that the probability of death follows a **binomial distribution**:

where the **logit transformation** is applied to mortality probabilities:

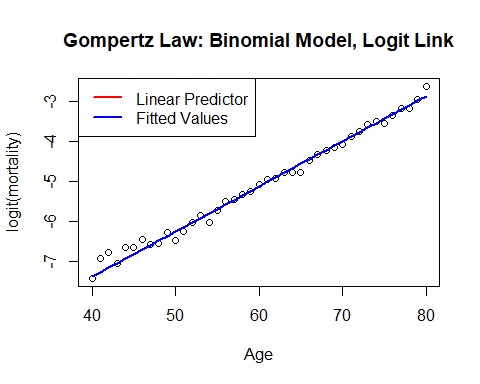
### Fit the Binomial Model

E.init <- EXP + DTH/2 # Initial exposure to risk  
Q.x <- DTH / E.init # Mortality probability estimate  
Logit <- function(x) log(x / (1 - x)) # Define logit function  
Obs <- Logit(Q.x) # Transform mortality rates  
  
# Fit the Binomial GLM  
Gomp.bin <- glm(Q.x ~ AGE, weights = E.init, family = binomial)  
summary(Gomp.bin)

Call:  
glm(formula = Q.x ~ AGE, family = binomial, weights = E.init)  
  
Deviance Residuals:   
 Min 1Q Median 3Q Max   
-3.2860 -0.5532 0.1287 1.0255 4.1660   
  
Coefficients:  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) -11.858352 0.073177 -162.05 <2e-16 \*\*\*  
AGE 0.112203 0.001141 98.34 <2e-16 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
(Dispersion parameter for binomial family taken to be 1)  
  
 Null deviance: 9483.109 on 40 degrees of freedom  
Residual deviance: 90.804 on 39 degrees of freedom  
AIC: 381.59  
  
Number of Fisher Scoring iterations: 4

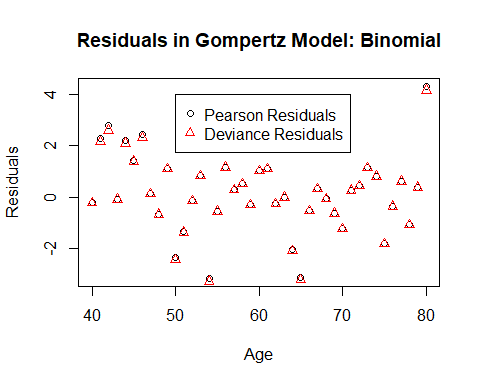
### Visualize the Binomial Model

plot(AGE, Obs, xlab = "Age", ylab = "logit(mortality)",  
 main = "Gompertz Law: Binomial Model, Logit Link")  
  
# Add the fitted Binomial regression line  
lines(AGE, Gomp.bin$coef[1] + Gomp.bin$coef[2] \* AGE, col = "red", lwd = 2)  
lines(AGE, Logit(Gomp.bin$fit), col = "blue", lwd = 2)  
  
legend("topleft", legend = c("Linear Predictor", "Fitted Values"),  
 col = c("red", "blue"), lty = c(1, 1), lwd = 2)



### Perform Residual Analysis for Binomial Model

plot(AGE, resid(Gomp.bin, type = "pear"), xlab = "Age",  
 ylab = "Residuals", main = "Residuals in Gompertz Model: Binomial")  
points(AGE, resid(Gomp.bin, type = "deviance"), pch = 2, col = "red")  
legend(50, 4, legend = c("Pearson Residuals", "Deviance Residuals"),   
 pch = c(1, 2), col = c("black", "red"))



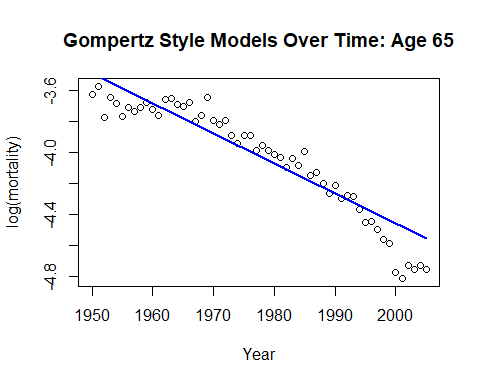
### Interpret the Residuals

* If residuals are **randomly distributed**, the model is a good fit.
* **Patterns in residuals** suggest a need for a more flexible model (e.g., a quadratic term or splines).
* Large residuals may indicate **outliers or data misfits**.

## Gompertz Models Over Time

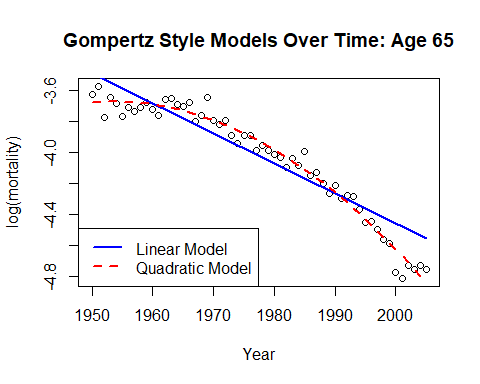
We now extend the Gompertz model to analyze how mortality changes over time for a given age.

DTH <- Dth[Age == 65, ]  
EXP <- Exp[Age == 65, ]  
Obs <- log(DTH / EXP)  
  
plot(Year, Obs, xlab = "Year", ylab = "log(mortality)",  
 main = "Gompertz Style Models Over Time: Age 65")  
  
Gomp.yr <- glm(DTH ~ offset(log(EXP)) + Year, family = poisson)  
lines(Year, log(Gomp.yr$fit / EXP), lwd = 2, col = "blue")



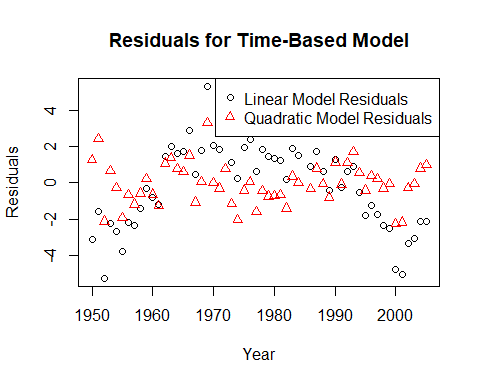
### Quadratic Model for Time Trends

Year2 <- (Year - mean(Year))^2  
Gomp.yr.2 <- glm(DTH ~ offset(log(EXP)) + Year + Year2, family = poisson)  
plot(Year, Obs, xlab = "Year", ylab = "log(mortality)",  
 main = "Gompertz Style Models Over Time: Age 65")  
lines(Year, log(Gomp.yr$fit / EXP), lwd = 2, col = "blue")  
  
lines(Year, log(Gomp.yr.2$fit / EXP), lwd = 2, lty = 2, col = "red")  
legend("bottomleft", legend = c("Linear Model", "Quadratic Model"),  
 col = c("blue", "red"), lty = c(1, 2), lwd = 2)



### Residual Analysis for Time Trends

plot(Year, resid(Gomp.yr, type = "pear"), xlab = "Year",  
 ylab = "Residuals", main = "Residuals for Time-Based Model")  
points(Year, resid(Gomp.yr.2, type = "pear"), pch = 2, col = "red")  
legend("topright", legend = c("Linear Model Residuals", "Quadratic Model Residuals"),  
 pch = c(1, 2), col = c("black", "red"))



## **Conclusion**

* The **Poisson model** works well but may suffer from overdispersion.
* The **Binomial model** provides an alternative framework based on survival probabilities.
* Time-based models suggest that mortality rates change over time and may require **nonlinear adjustments**.