Tutorial 5: Graduation I

Yubo Rasmussen

## Tutorial 5: Graduation I

### Question 1

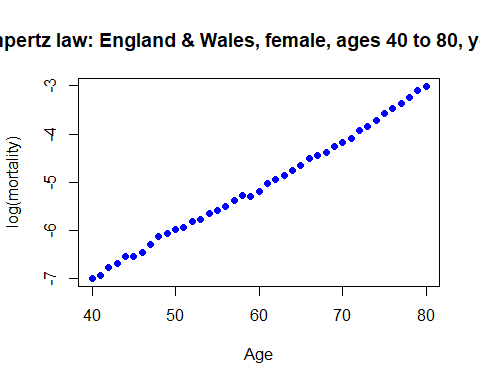
Extract the data for females ages 40 to 80 for the year 2005 from the England and Wales (E&W) data set. Use R to fit Gompertz type models; i.e., for assume that is linear in age while for assume that is linear in .

#### Load the Data

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

#### Extract the Data

#  
# Extract female data, ages 40 to 80, year 2005.  
#  
DTH <- Dth.F[ (40 <= Age) & (Age <= 80), Year == 2005]  
EXP <- Exp.F[ (40 <= Age) & (Age <= 80), Year == 2005]  
AGE <- 40:80  
Obs <- log(DTH/EXP)  
plot(AGE, Obs, xlab = "Age", ylab = "log(mortality)",  
 main = "Gompertz law: England & Wales, female, ages 40 to 80, year 2005",  
 pch = 16, col = "blue")



#  
# It will be useful to have this as a function so  
#  
Main.plot <- function( ) plot(AGE, Obs, xlab = "Age", ylab = "log(mortality)",  
 main = "Gompertz law: England & Wales, female, ages 40 to 80, year 2005",  
 pch = 16, col = "blue")

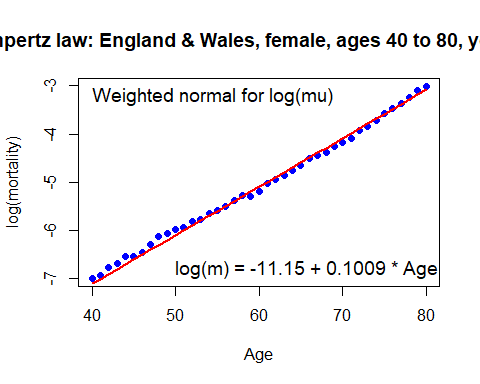
#### Part (a)

The model for using weighted least squares to estimate the parameters.

#  
# (a) Weighted normal for mu\_x  
#  
Gomp.wt <- lm(Obs ~ AGE, weights = DTH)  
Gomp.wt$coef

(Intercept) AGE   
-11.1509845 0.1008921

Main.plot( )  
lines(AGE, Gomp.wt$fit, lwd = 2, col = "red")  
text(50, -6.8, "log(m) = -11.15 + 0.1009 \* Age", adj = 0, cex = 1.2)  
text(40, -3.2, "Weighted normal for log(mu)", adj = 0, cex = 1.2)



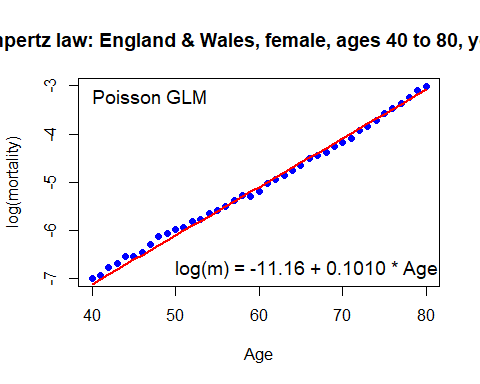
#### Part (b)

The model for using a GLM with Poisson errors to estimate the parameters.

#  
# (b) GLM with Poisson errors  
#  
OFF <- log(EXP)  
Gomp.P <- glm(DTH ~ offset(OFF) + AGE, family = poisson)  
Gomp.P$coef

(Intercept) AGE   
-11.1601025 0.1009924

Main.plot( )  
lines(AGE, Gomp.P$lin - OFF, lwd = 2, col = "red")  
text(50, -6.8, "log(m) = -11.16 + 0.1010 \* Age", adj = 0, cex = 1.2)  
text(40, -3.2, "Poisson GLM", adj = 0, cex = 1.2)



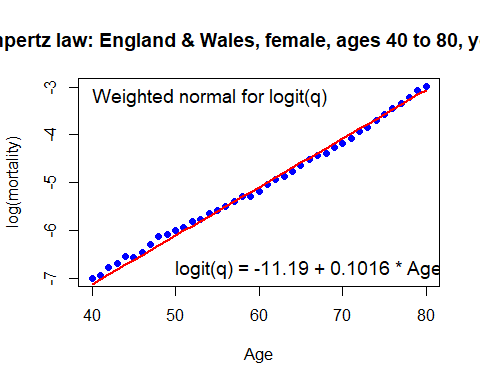
#### Part (c)

The model for using weighted least squares to estimate the parameters. (Hint: You need ).

#  
# (c) Weighted normal for q\_x  
#  
Logit <- function(x) log(x/(1-x))  
E.init <- EXP + 0.5\*DTH  
Q.x <- DTH/E.init  
Obs <- Logit(Q.x)  
Gomp.wt.Q <- lm(Obs ~ AGE, weight = DTH)  
Gomp.wt.Q$coef

(Intercept) AGE   
-11.1893597 0.1016086

Main.plot( )  
lines(AGE, Gomp.wt.Q$fit, lwd = 2, col = "red")  
text(50, -6.8, "logit(q) = -11.19 + 0.1016 \* Age", adj = 0, cex = 1.2)  
text(40, -3.2, "Weighted normal for logit(q)", adj = 0, cex = 1.2)



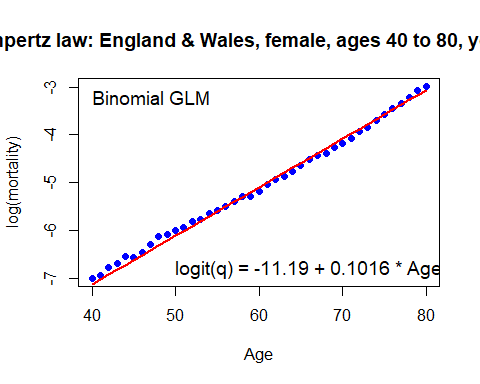
#### Part (d)

The model for using a GLM with binomial errors to estimate the parameters.

#  
# (d) GLM with binomial errors  
#  
Gomp.B <- glm(Q.x ~ AGE, family = binomial, weight = E.init)  
Gomp.B$coef

(Intercept) AGE   
-11.1940994 0.1016374

Main.plot( )  
lines(AGE, Gomp.B$lin, lwd = 2, col = "red")  
text(50, -6.8, "logit(q) = -11.19 + 0.1016 \* Age", adj = 0, cex = 1.2)  
text(40, -3.2, "Binomial GLM", adj = 0, cex = 1.2)



#### Part (e)

Compare the GLM graduations of and .

#  
# (e) Clearly all four fits are very similar  
#

#### Part (f)

Obtain the standardised residuals, , in each of the four graduations and plot the against . Comment informally on what these plots tell you.

#  
# (f)  
#  
# Normal residuals  
#  
Obs <- log(DTH/EXP)  
summary(Gomp.wt)

Call:  
lm(formula = Obs ~ AGE, weights = DTH)  
  
Weighted Residuals:  
 Min 1Q Median 3Q Max   
-6.1260 -2.9005 0.6493 2.8098 7.0446   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) -11.150985 0.076580 -145.61 <2e-16 \*\*\*  
AGE 0.100892 0.001093 92.31 <2e-16 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 3.459 on 39 degrees of freedom  
Multiple R-squared: 0.9954, Adjusted R-squared: 0.9953   
F-statistic: 8521 on 1 and 39 DF, p-value: < 2.2e-16

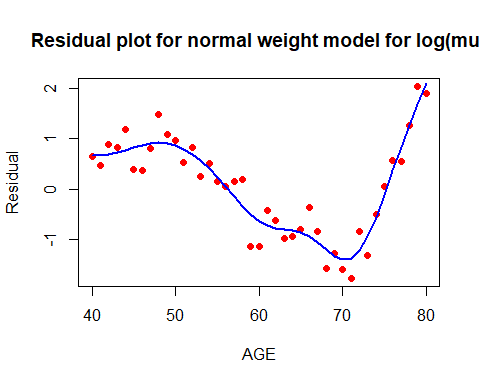
SE <- 3.459  
Z.a <- (Obs - Gomp.wt$fit)/SE\*sqrt(DTH)  
plot(AGE, Z.a, ylab = "Residual", col = "red", pch = 16,  
 main = "Residual plot for normal weight model for log(mu)")  
#  
# As an extra, here is how to fit a curve through these points  
#  
library(mgcv)

Warning: package 'mgcv' was built under R version 4.2.3

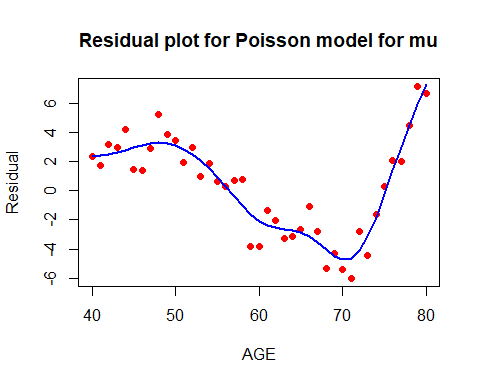
Loading required package: nlme

This is mgcv 1.9-0. For overview type 'help("mgcv-package")'.

lines(AGE, gam(Z.a ~ s(AGE))$fitted, col = "blue", lwd = 2)



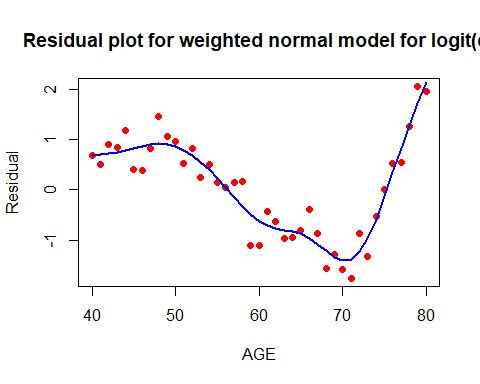
#  
# Poisson residuals  
#  
Z.b <- (DTH - Gomp.P$fit)/sqrt(Gomp.P$fit)  
plot(AGE, Z.b, ylab = "Residual", col = "red", pch = 16,  
 main = "Residual plot for Poisson model for mu")  
lines(AGE, gam(Z.b ~ s(AGE))$fitted, col = "blue", lwd = 2)



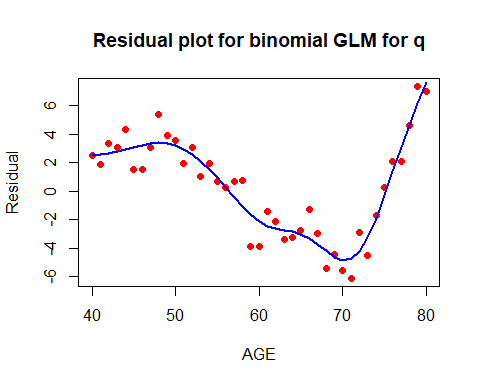
#  
# Binomial residuals: weighted normal fit  
#  
Obs <- Logit(Q.x)  
summary(Gomp.wt.Q)

Call:  
lm(formula = Obs ~ AGE, weights = DTH)  
  
Weighted Residuals:  
 Min 1Q Median 3Q Max   
-6.3749 -3.1137 0.6215 2.9542 7.4512   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) -11.189360 0.080404 -139.16 <2e-16 \*\*\*  
AGE 0.101609 0.001148 88.54 <2e-16 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 3.632 on 39 degrees of freedom  
Multiple R-squared: 0.995, Adjusted R-squared: 0.9949   
F-statistic: 7840 on 1 and 39 DF, p-value: < 2.2e-16

SE <- 3.632  
Z.c <- (Obs - Gomp.wt.Q$fit)/SE\*sqrt(DTH)  
plot(AGE, Z.c, ylab = "Residual", col = "red", pch = 16,  
 main = "Residual plot for weighted normal model for logit(q)")  
lines(AGE, gam(Z.c ~ s(AGE))$fitted, col = "blue", lwd = 2)



#  
# Binomial GLM  
#  
DTH.dot <- E.init \* Gomp.B$fit  
Z.d <- (DTH - DTH.dot)/sqrt(DTH.dot)  
plot(AGE, Z.d, ylab = "Residual", col = "red", pch = 16,  
 main = "Residual plot for binomial GLM for q")  
lines(AGE, gam(Z.d ~ s(AGE))$fitted, col = "blue", lwd = 2)



#  
# COMMENT: The residuals for the normal models are fine in size but  
# have an obvious pattern. The residuals for the Poisson and  
# binomial models have the same pattern but are much larger; this  
# is because the normal models have a separate variance parameter  
# which injects more (and needed!) variation into the system. This  
# can be dealt with but goes beyond the course.  
#

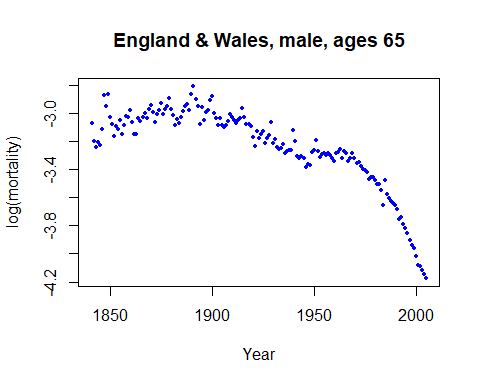
### Question 2

#### Part (a)

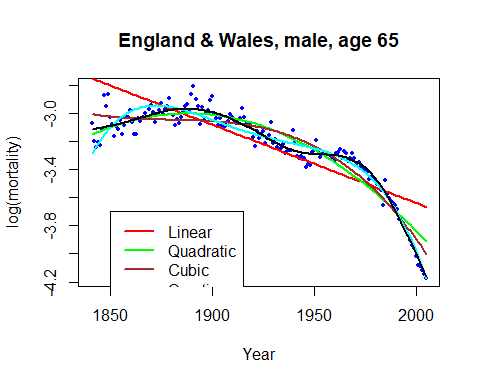
Extract the male data for age 65 for all years from the E&W data set. Plot the maximum likelihood estimates of against year, . Consider Gompertz style graduations of with

for Use the glm( ) function and Poisson errors to estimate the regression coefficients. Compare the various fits with appropriate plots. *Hint: You will do well to use powers of* *in the calls of glm( ); see section 5.1.4.*

#  
# Extract female data, age 65, years 1841 to 2005 (all years).  
#  
DTH <- Dth.M[ Age == 65, ]  
DTH <- round(DTH)  
EXP <- Exp.M[ Age == 65, ]  
Obs <- log(DTH/EXP)  
plot(Year, Obs, ylab = "log(mortality)", cex = 0.5,  
 main = "England & Wales, male, ages 65", pch = 16, col = "blue")



#  
# It will be useful to have this as a function so  
#  
Main.plot <- function( ) plot(Year, Obs, ylab = "log(mortality)", cex = 0.5,  
 main = "England & Wales, male, age 65", pch = 16, col = "blue")  
  
#  
# Define some powers of year  
#  
Mean <- mean(Year)  
Y2 <- (Year - Mean)^2  
Y3 <- (Year - Mean)^3  
Y4 <- (Year - Mean)^4  
OFF <- log(EXP)  
#  
Model.1 <- glm(DTH ~ offset(OFF) + Year, family = poisson)  
Main.plot( )  
lines(Year, Model.1$lin - OFF, lwd = 2, col = "red")  
#  
Model.2 <- glm(DTH ~ offset(OFF) + Year + Y2, family = poisson)  
lines(Year, Model.2$lin - OFF, lwd = 2, col = "green")  
#  
Model.3 <- glm(DTH ~ offset(OFF) + Year + Y2 + Y3, family = poisson)  
lines(Year, Model.3$lin - OFF, lwd = 2, col = "brown")  
#  
Model.4 <- glm(DTH ~ offset(OFF) + Year + Y2 + Y3 + Y4, family = poisson)  
lines(Year, Model.4$lin - OFF, lwd = 2, col = "cyan")  
#  
# and for reference here is a general smooth curve  
#  
library(mgcv)  
Smooth.Fit <- log(gam(DTH ~ offset(OFF) + s(Year), family = poisson)$fitted/EXP)  
lines(Year, Smooth.Fit, lwd = 2, col = "black")  
#  
# Add a legend  
#  
legend(1850, -3.7, legend = c("Linear", "Quadratic", "Cubic", "Quartic",  
 "Smooth"), lty = 1, lwd = 2,  
 col = c("red", "green", "brown", "cyan", "black"))



#### Part (b)

Repeat (a) for age 20.

#  
# Age 20 can be done with a change of data and the same program.  
# The labelling needs altered (although this too can be made automatic).  
#

#### Part (c)

Note for interest the amazing changes in mortality that have taken place from 1841 to 2005.