Assignment 2 Masskill

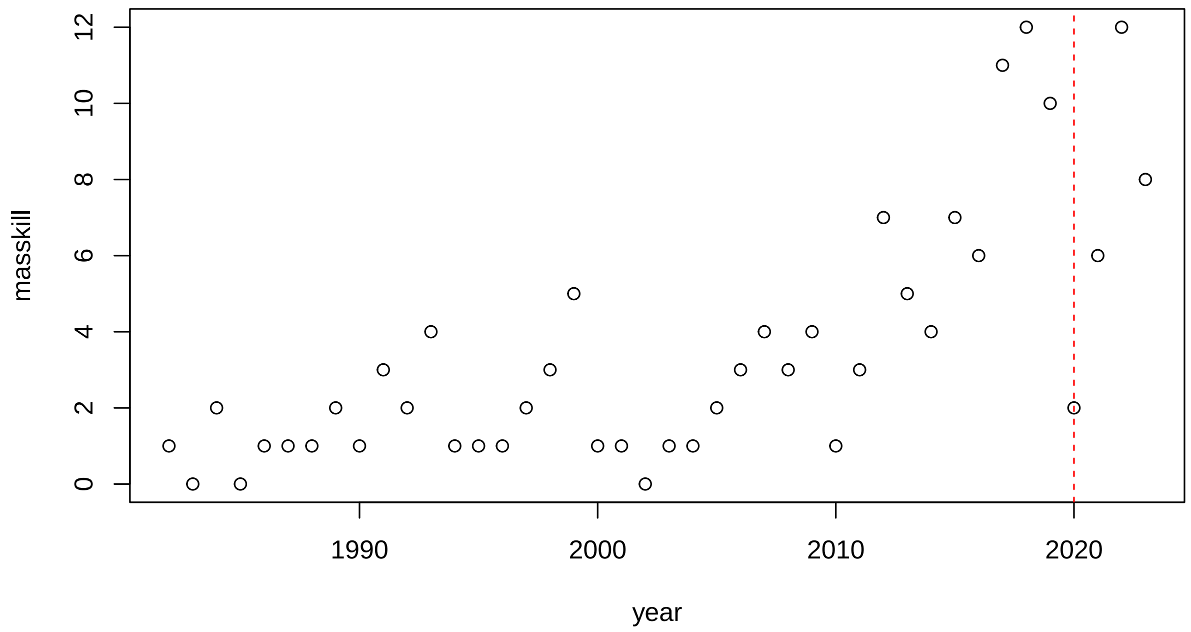
Ben Lu

2023-08-11

Question 1

(a) Mass killings per year

```r
masskill.df <- read.csv("masskill.csv")
plot(masskill ~ year, data = masskill.df)
abline(v = 2020, col = "red", lty = 2)
```

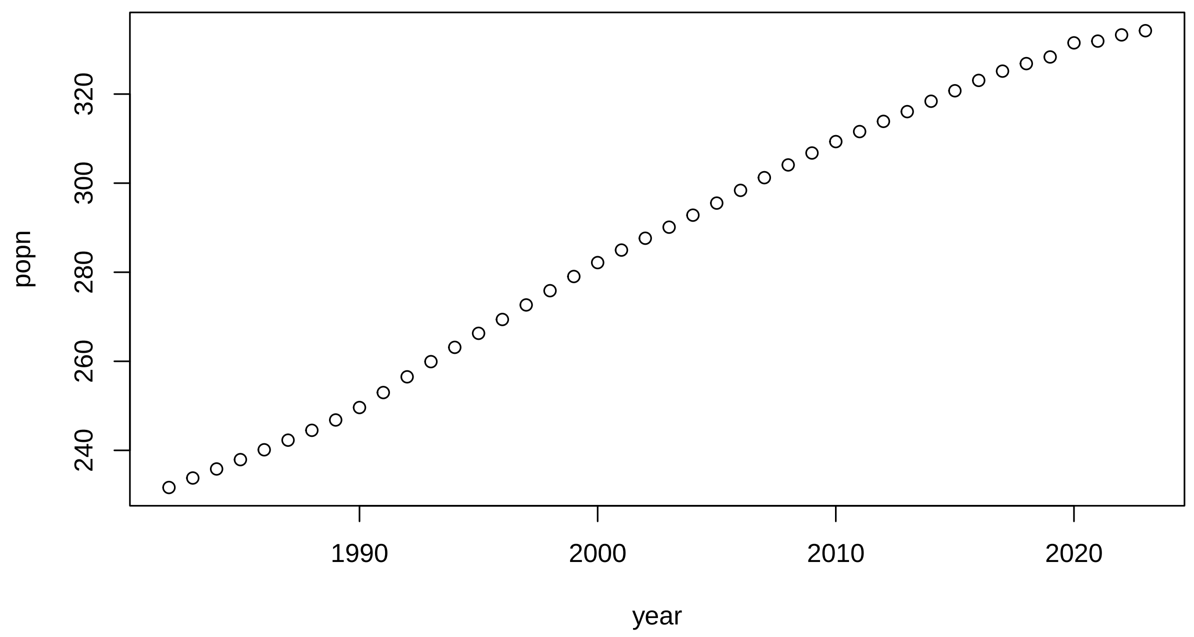


In our plotted data, there appears to be a steady increase in the number of mass killing at the start, but the number of mass killing increases rapidly as year increases. This suggests an exponential relationship between mass killing and the year which it occurred.

In the period of time from 1982 to 2023, it seems the number of mass killing increased from approximately 2 to 12, which is a six-fold increase in frequency. There is also an outlier in our data for the mass killing in 2020, which dropped significantly as a result of the lockdown.

(b) Plot of US population per year

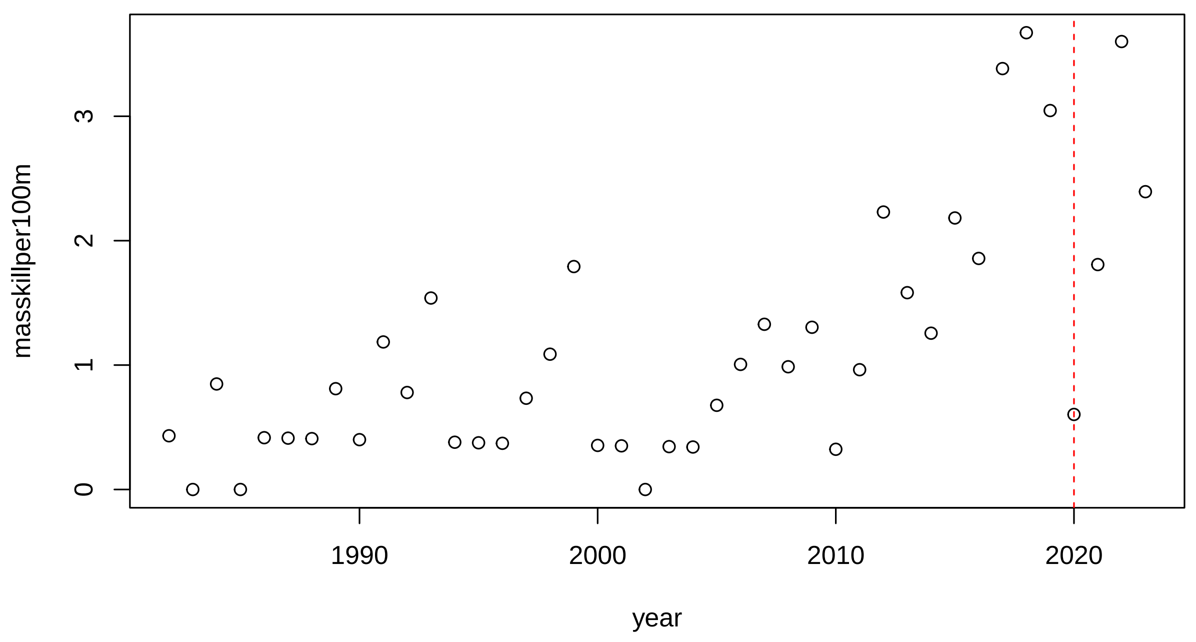
```r
plot(popn ~ year, data = masskill.df)
```



It appears that the US population steadily increases throughout the period of time, the relationship is approximately linear. There is a positive linear relationship between US population and year.

(c) Mass killing per year, per 100 million people.

```r
masskill.df$masskillper100m <- (masskill.df$masskill/masskill.df$popn) * 100
plot(masskillper100m ~ year, data = masskill.df)
abline(v = 2020, col = "red", lty = 2)
```



The number of mass killing per 100 million people increases steadily at the start, but the rate of killing increases rapidly as the number of year increases, which suggests an exponential relationship between mass killing per 100 million people and year. Apart from the outlier in our data during the 2020 lockdown, it appears the rate of the events has gone from around one mass killing per 100 million people to around three mass killing per 100 million people. There has been an at least threefold increase in the rate of occurrence throughout 1982 to 2023.

(d)

Each mass killing in our data is treated independently as each perpetrator who carry out the events are acting on their own without the influence of other perpetrators.

(e) Fitting GLM for mass killing

```
MK.fit=glm(masskill ~ I(year - 1982)*(year>=2020),
subset = (year!=2023),family = "poisson",
offset = log(popn/100),data = masskill.df )
anova(MK.fit,test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: masskill
##
## Terms added sequentially (first to last)
```

```
##
##
##                               Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                            40     92.927
## I(year - 1982)                 1   47.373        39     45.554 5.869e-12 ***
## year >= 2020                   1    1.927        38     43.627  0.165091
## I(year - 1982):year >= 2020    1    6.672        37     36.955  0.009791 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# glm(masskill ~ I(year - 1982)*(year>=2020) creates a linear regression model
# of masskill against year.

# I(year - 1982) are years minus 1982 which plots the mass killing against the
# number of years from 1982 on-wards.

#*(year>=2020) creates an indicator variable which is equal to 1 if
#during/after lockdown in 2020, and equal 0 otherwise.

# subset = (year!=2023),family = "poisson" : using subset excludes the year of
# 2023 from the analysis and set GLM to poisson distribution.

# offset = log(popn/100),data = masskill.df ) : using offset will add
# log(popn/100) to our equation, this accounts for the effect of the increasing
# US population on the number of mass killing by adding the offset.

# anova(MK.fit,test = "Chisq") : creates Chi-squared test statistic
```

(f) Mathematical model

Our mathematical model is:

$$log(E(count_i)) = \beta_0 + \beta_1 \times (year_i - 1982) + \beta_2 \times postLockdown_i + \beta_3 \times (year_i - 1982) \times postLockdown_i + log(popn_i/100)$$

where $E(count_i)$ is the mass killing count, $year_i$ is the number of years, $postLockdown_i$ is an indicator variable which takes the value 1 if the year is 2020 or greater to account for the effects of after the lockdown, $popn_i$ is the US population in millions which accounts for the offset due to that year's population.

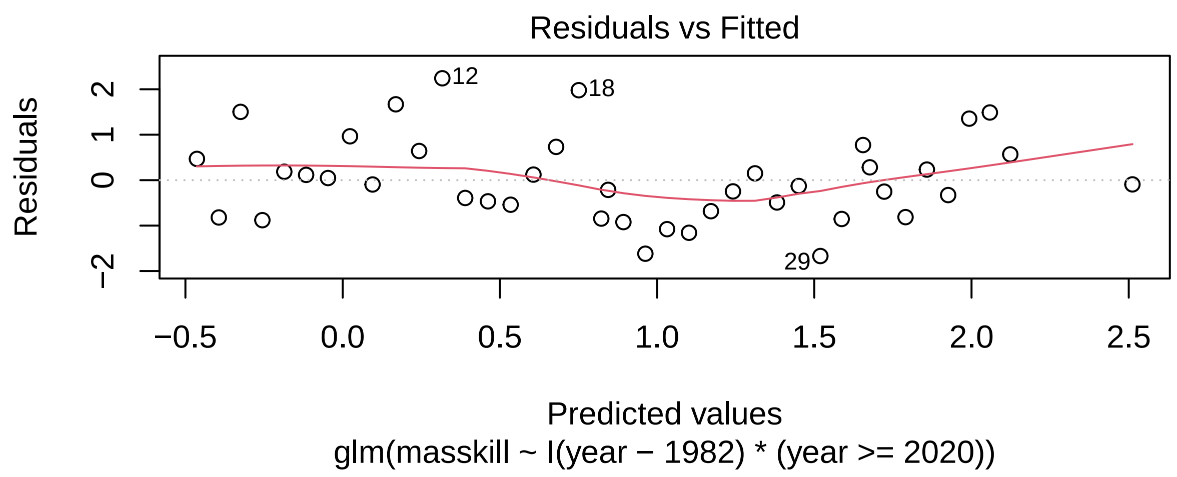
(g) Summary

```
summary(MK.fit)
```

```
##
## Call:
## glm(formula = masskill ~ I(year - 1982) * (year >= 2020), family = "poisson",
##     data = masskill.df, subset = (year != 2023), offset = log(popn/100))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.28858  -0.73458  -0.09668   0.55041   1.81664
##
```

```
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -1.303519   0.274980  -4.740 2.13e-06 ***
## I(year - 1982)                  0.060486   0.009941   6.085 1.17e-09 ***
## year >= 2020TRUE              -30.628662  12.718475  -2.408   0.0160 *
## I(year - 1982):year >= 2020TRUE   0.770516   0.322015   2.393   0.0167 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 92.927  on 40  degrees of freedom
## Residual deviance: 36.955  on 37  degrees of freedom
## AIC: 153.63
##
## Number of Fisher Scoring iterations: 5
```

```r
plot(MK.fit, which = 1)
```



```r
pvalue = 1 - pchisq(deviance(MK.fit), df.residual(MK.fit))
pvalue
```

```
## [1] 0.471162
```

We obtained a large p-value of 0.47 in our check for over-dispersion which indicates no evidence for over-dispersion and that our GLM fit using Poisson distribution was adequate.

There is a slight curvature in our residuals plot, but it doesn't indicate much of a concern.

As we expect a greater population would have more mass killing, the offset term accounts for the impact of the increasing US population on the mass killing by adjusting for the annual population change. This means the increasing US population has no impact in the mass killing count for our analysis.

Regardless of the increase in US population, it appears that for every increase in year from 1982 onward, the log count of the mass killing increases by 0.06.

During and after the lockdown in 2020, the log count of the mass killing decreased by 30.63. Also for every year increase after the lockdown, the log count of mass killing increases by 0.83 (0.0605 + 0.7705).

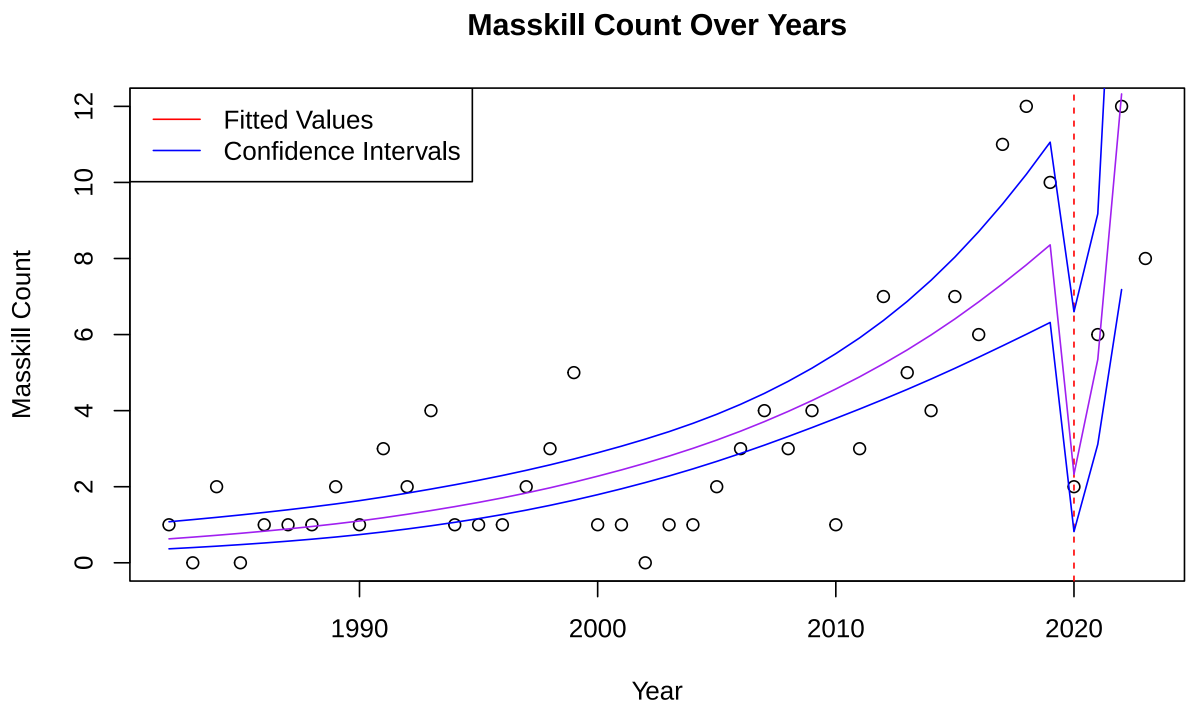
(h) Confidence intervals for plot of Masskill and Years

```
preds=predict(MK.fit, interval = "confidence", se.fit=TRUE) # Generate confidence values
year = masskill.df$year[-42]

plot(masskill ~ year, data = masskill.df, xlab = "Year", ylab = "Masskill Count", main = "Masskill Coun
abline(v = 2020, col = "red", lty = 2)

# Superimpose the fitted values and prediction intervals on the plot
lines(year, exp(preds[["fit"]]), col = "purple", lty = 1)  # Fitted values
lines(year, exp(preds[["fit"]] - 1.96 * preds[["se.fit"]]), col = "blue", lty = 1) # Lower bound
lines(year, exp(preds[["fit"]] + 1.96 * preds[["se.fit"]]), col = "blue", lty = 1) # Upper bound

# Add a legend to the plot
legend("topleft", legend = c("Fitted Values", "Confidence Intervals"), col = c("red", "blue"), lty = c(
```



## (i) Confidence intervals for 2023

```
predict.2023 <- data.frame(year = 2023, popn = 334.23)
confidence <- predict(MK.fit, newdata = predict.2023, interval = "confidence", se.fit = TRUE)
mean <- confidence$fit #log mean count
se <- confidence$se.fit #log standard error

# mean point estimate
exp(mean)
```

```
##        1
## 28.37673
```

```r
# upper confidence bound
exp(mean + 1.96 *se)
```

```
##        1
## 80.51216
```

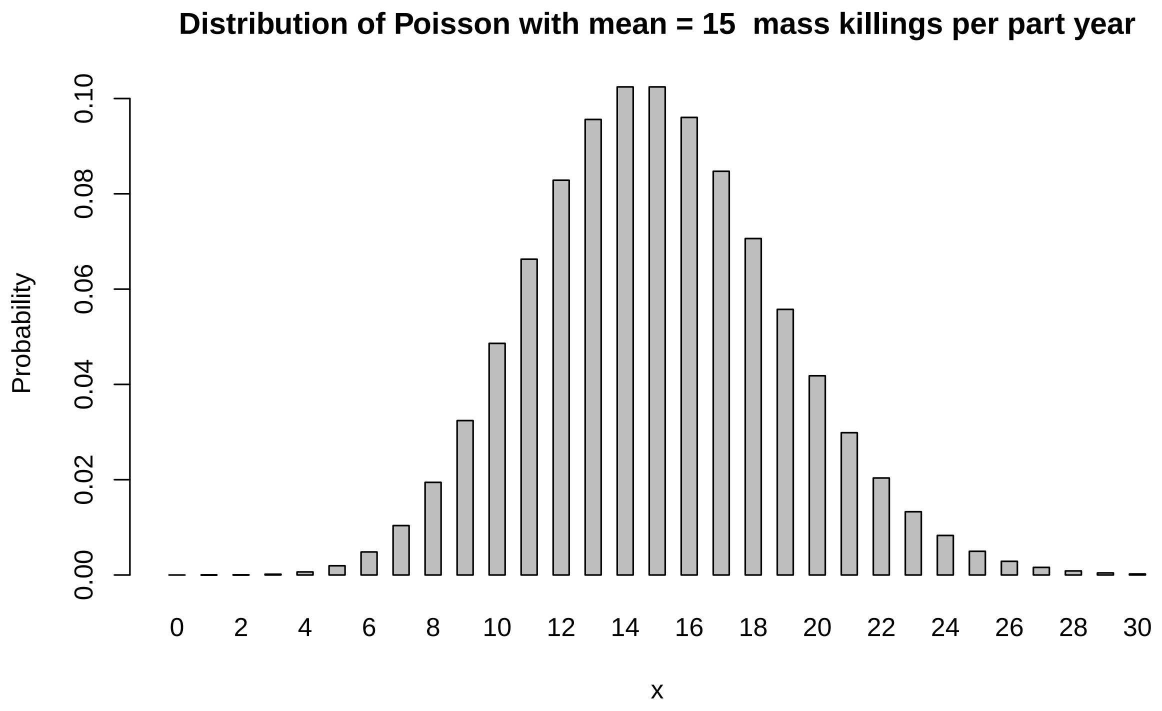
```r
# lower confidence bound
exp(mean - 1.96 *se)
```

```
##        1
## 10.00145
```

Under the assumption that the US population in 2023 is 334.23 million, we estimate with 95% confidence that the mean number of mass killing is between 10 and 80.5 counts. The mean point estimate of our model is 28.38 mass killing.

(j)

```r
# distribution of Poisson(lambda=mean) distribution
pred.mean=15 # change this
barplot(dpois(0:30,pred.mean),
ylab="Probability",xlab="x",
space=1, names.arg=0:30,
main=paste("Distribution of Poisson with mean =",
round(pred.mean,2)," mass killings per part year"))
```



```r
# hypothesis test
cdf = 1-ppois(7,15)
cdf
```

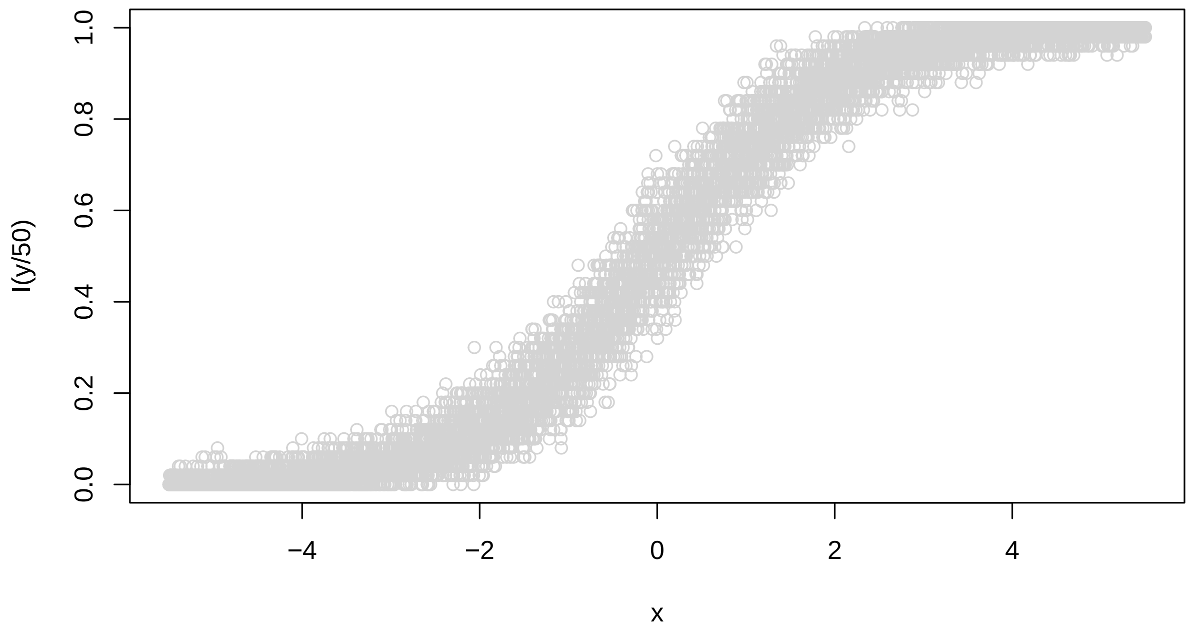
```
## [1] 0.9819978
```

As of July 12th 2023, we expect the number of mass killing to be 15 (193/365 * 28.38) according to our model estimate. Under the assumption that our model estimate of 15 mass killing was appropriate, we observe that there is a large p-value of 0.98 probability that 8 or more mass killing occurred, in which we have no evidence against our hypothesis of 15 mass killing.

As we have no evidence against our null hypothesis, we are confident in the accuracy of our model.

Question 2

```r
set.seed(345)
nx=1e4
x=seq(-5.5, 5.5, length=nx)
y=rbinom(n=nx,size=50, prob=exp(0+1*x)/(1+exp(0+1*x)))
plot(I(y/50)~x, col="lightgrey")
```



```r
mod=glm(cbind(y,50-y)~x, family=binomial)
```

(a)

```r
l = sum(log(dbinom(y, size = 50, prob = fitted(mod))))
l.sat = sum(log(dbinom(y, size = 50, prob = y/50)))
l.null = sum(log(dbinom(y, size = 50, prob = mean(y)/50)))

# null deviance, manual calculation
null.deviance = 2 * (l.sat - l.null)
null.deviance
```

```
## [1] 409351
```

```r
# null deviance, r calculation
mod$null.deviance
```

```
## [1] 409351
```

```r
# residual deviance, manual calculation
resid.deviance = 2 * (l.sat - l)
resid.deviance
```

```
## [1] 10120.79
```

```r
# residual deviance, r calculation
mod$deviance
```

```
## [1] 10120.79
```

(b)

```r
b.l = log(dbinom(y, size = 50, prob = fitted(mod)))
b.l.sat = log(dbinom(y, size = 50, prob = y/50))
b.l.null = log(dbinom(y, size = 50, prob = mean(y)/50))

# manual calculation of residual deviance
manual.resid.dev = sqrt(2*(b.l.sat - b.l)) * ifelse(y/50<fitted(mod),-1,1)
# r computing residual deviance
r.resid.dev = resid(mod, "deviance")

# range of the differences between manual and r calculation for all values
# of residual deviance. The range is very tiny, which means the manual
# calculations are very accurate.
range(manual.resid.dev - r.resid.dev)
```

```
## [1] -1.319116e-11  9.580015e-12
```

(c)

```r
c.l = log(dbinom(y, size = 50, prob = fitted(mod)))
c.l.sat = log(dbinom(y, size = 50, prob = y/50))
c.l.null = log(dbinom(y, size = 50, prob = mean(y)/50))

# pearson residuals
pearson.resid = (y-(50*fitted(mod)))/sqrt((50*fitted(mod))*(1-fitted(mod)))
# compare our manual calculations for pearson residual with r calculation
range(pearson.resid - resid(mod,type = "pearson"))
```

```
## [1] -7.771561e-15  9.325873e-15
```

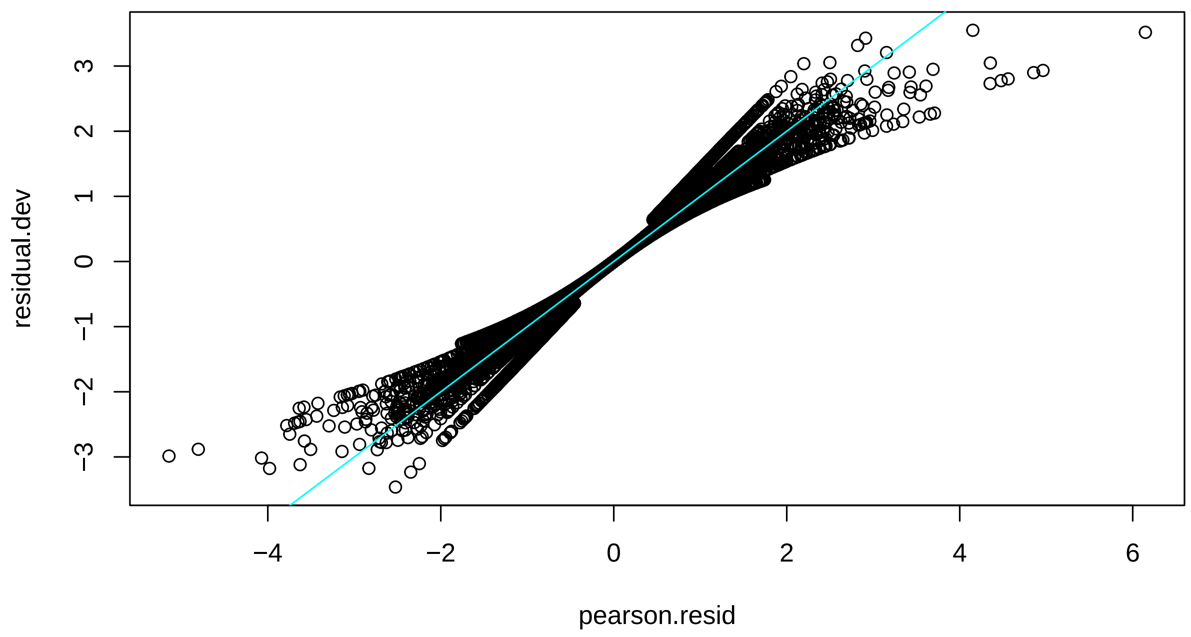
```r
# residual deviance
residual.dev = sqrt(2*(c.l.sat - c.l)) * ifelse(y/50<fitted(mod),-1,1)
# compare our manual residual deviance with r calculation
range(residual.dev-resid(mod, "deviance"))
```

```
## [1] -1.319116e-11  9.580015e-12
```

(d)

```
# the plot of residual deviance and pearson residuals is approximately
# a relationship with a slope of 1, which means our model residual
# deviance are approximately the same as our pearson residuals.

plot(residual.dev ~ pearson.resid)
abline(a=0, b=1, col="cyan")
```


residual.dev
pearson.resid