BEE552 Biometry Week 10

Maria Feiler

04/06/2022

My Learning Journey

Over the last week, I participated in Biometry in the following ways:

- I asked / answered **3** questions posed in class.
- I asked **1** questions in Slack.
- I answered **0** questions posed by other students on Slack.
- I came to Heather's office hours: **Yes**
- I came to Jose's office hours: **No**
- I met with Heather or Jose separately from office hours: **No**

Anything not falling into one of the above categories?

> **No**

On a scale of 1 (no knowledge) to 10 (complete expert), how would I rate my comfort with R programming after this week?

> **7**

Any topics from last week that you are still confused about?

> More of a coding thing than a math thing, but I was struggling to find certain values within the glm() objects and the like. How would I find that median that I used as a start value vs just copying and pasting the numerical value?
>
> Also, since we're going to be needing it soon, how on earth do nested for-loops work. I truely cannot wrap my head around it.

Problem Set

Part I

Fill out the following table.

Test	H_0	Test Statistic T	$f(T \mid H_0)$	Assumptions
Pearson's product moment correlation	$\rho = 0$	$T^* = r\sqrt{\frac{n-2}{1-r^2}}$	t_{n-2}	• Joint probability (A,B) is bivariate normal • Relationship between A and B is linear • Data are independent samples from a joint distribution
Spearman's rank correlation	$\rho = 0$	$r_s = \frac{Cov(ranks_A, ranks_B)}{\sqrt{Var(ranks_A)Var(ranks)}}$	$\sqrt{\frac{1}{n-1}}N(0,1)$	• Data are independent samples from a joint distribution
Kendall's coefficient of rank correlation	$\tau = 0$	$\tau = \frac{concordant\ pairs - discordant\ pairs}{0.5n(n-1)}$	$N\left(0, \frac{2(2n+5)}{9n(n-1)}\right)$	• Data are continuous and ordinal • Data are monotonic

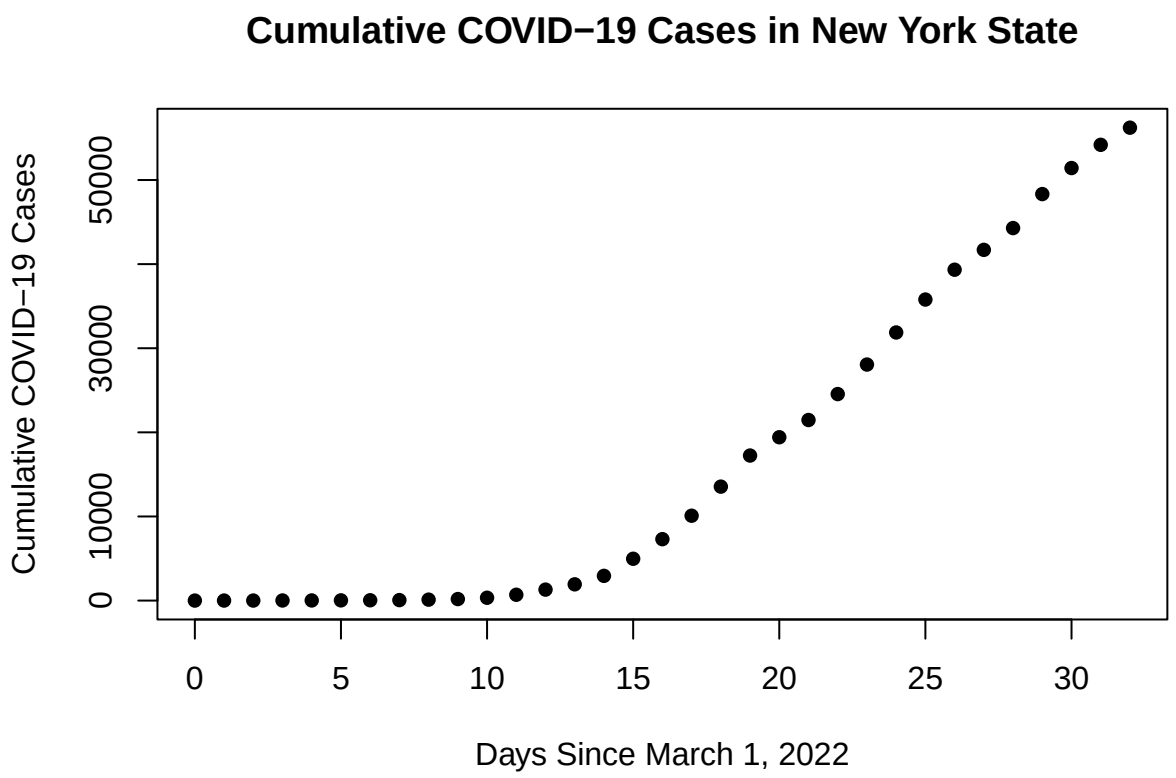
where $Var(A) = \frac{1}{n-1}\sum_{i=1}^{n}(A_i - \bar{A})(A_i - \bar{A})$ and $Cov(A,B) = \frac{1}{n-1}\sum_{i=1}^{n}(A_i - \bar{A})(B_i - \bar{B})$

Part II

Load the dataset "NYData.csv". These data represent number of new confirmed coronavirus cases in the first days after March 1, 2020 in the state of New York. (March 1st is Day=0.) There are 19.54 million residents in the state of New York.

```
nyData <- readxl::read_xls("NYData updated 2022.xls", sheet = 1)

# Add cumulative sum column
nyData$CaseCumSum <- cumsum(nyData$NewCases)

# Add a column for those not infected
nyData$Uninfected <- 19540000-nyData$CaseCumSum

# Remove February 29 data ("Day 0") and reset March 1 to Day 0
nyData <- nyData[-1,]
nyData$Day <- nyData$Day-1
```

a. Plot the number of cumulative cases as a function of Day.



b. Fit the cumulative coronavirus data Y using logistic regression, and plot the best fit line on the original data. Using R's function 'predict.glm', plot the 95th percentile confidence intervals as well (we can assume the errors are normally distributed and use ±1.96 s.e.). Note that a logistic regression is used when the underlying data follow a Binomial process. Think carefully about how to set up your logistic regression so it reflects the Binomial nature of the data you have.

```
# Get logistic regression
fit1 <- glm(cbind(nyData$CaseCumSum, nyData$Uninfected) ~ nyData$Day,
              family = "binomial")

summary(fit1)
```

```
Call:
glm(formula = cbind(nyData$CaseCumSum, nyData$Uninfected) ~ nyData$Day,
    family = "binomial")

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-82.90  -49.80  -34.81   38.25   53.69

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.033e+01  5.935e-03 -1739.9   <2e-16 ***
nyData$Day   1.503e-01  2.219e-04   677.3   <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: 785245  on 32  degrees of freedom
Residual deviance:  69366  on 31  degrees of freedom
AIC: 69673

Number of Fisher Scoring iterations: 5
```

```r
# Create dataframe of days since March 1 (0 to 32)
newData <- data.frame(Day = seq(0,32))

# Calculate confidence interval bands
nyCIBands <- predict.glm(fit1,
                         newData,
                         se.fit = TRUE)
```

c. We will now fit an alternative model to these data, one that is often used for processes like epidemic spread. Fit the Gompertz curve $Y \sim N(ab^{be^{cx}}, \sigma^2)$*, which is equivalent to* $Y_i = ae^{be^{cx}} + \epsilon_i, \epsilon_i \sim N(0, \sigma^2)$*, to the coronavirus data. You might start trying R's simple function 'nls' but you might find that it is hard to find starting values that are close enough for 'nls' to actually converge on estimates. One trick is to take the log of the above equation (log both the left and right hand sides) and then try fitting that logged equation. This logged version is very similar (but not exactly the same) to the original question, and the estimates for this logged version will be good starting values. (Taking the log of both sides yields an equation which is numerically easier for 'nls' to handle.) If this doesn't work for you, you might also try the 'nlsLM' function in the 'minpack.lm' package. More details here: http://www.r-bloggers.com/a-better-nls/.*

```r
# Initial values were taken from the glm() results from before (chose these values
# after working with Emily and Sydney)

# The total number of cases 33 days after March 1
aStart <- nyData$CaseCumSum[33]

# The median of the deviance residuals
bStart <- -34.81

# The intercept
cStart <- -fit1$coefficients[[2]]

# Get Gompertz curve
fit2 <- nls(CaseCumSum ~ a*exp(b*exp(c*Day)),
            data = nyData,
            start = list(a = aStart,
                         b = bStart,
                         c = cStart)
           )
```

Write a script to construct the 95th percentile confidence intervals using the following bootstrap technique (this is a bit of a hack, but one is often left with no choice but to get creative...):

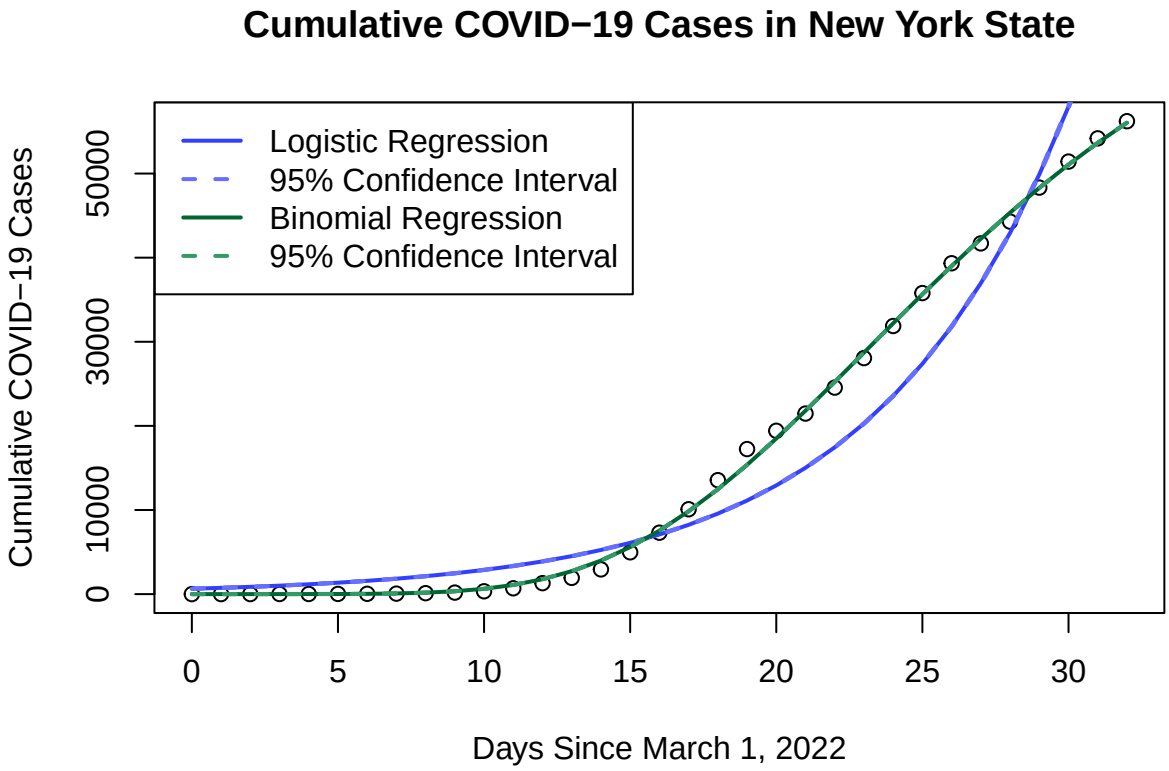
1. *sample with replacement from the residuals of the original 'nls' fit*
2. *construct a new bootstrap dataset using* $Y_i^* = \hat{Y}_i + \epsilon_i^*$ *" (Note that* ϵ_i^* *is a legitimate residual from the original fit, its just been moved to a new data point.)*
3. *refit the model as above*
4. *use nls() to obtain estimates for the model parameters (k is an index for which bootstrap sample you are on)* (a_k^*, b_k^*, c_k^*) *using the data created by the bootstrapping procedure*
5. *save the model predictions for each x (i.e.* $a_k^* e^{b_k^* e^(c_k^* x}$
6. *repeat steps 1-5 1000 times (this will generate 1000 bootstrap replications)*
7. *for each value of x (i.e. Day), define the 95th percentile CI for* $ae^{be^{cx}}$ *(at that x) as the (2.5th, 97.5th) percentiles of the bootstrapped* $ae^{be^{cx}}$ *(at that x)*

(Notice that your CI for $ae^{be}{cx}} will no longer be a smooth function! Also, you might find that inside your bootstrap loop, you are getting errors and this is causing the loop to abort with an error. This is happening because there will be times when, by random chance, the bootstrapped datasets will be hard to fit, and nls() will fail to converge. To avoid having an error on one iteration cause the whole loop to abort with an error, wrap the entire line of code with nls in the try() function. This will prevent an error from causing the loop to abort with an error. For more details, look at the help file ?try.)

```
# Define number of iterations
nit <- 1000

# Define the matrices into which residual bootstrap samples and the nls() results
# will be stored
boot <- matrix(data = NA,
               nrow = nit,
               ncol = length(fit1$residuals),
               dimnames = list(c(1:nit),
                               c(1:length(fit1$residuals))
                               )
               )

abcEst<- matrix(data = NA,
                nrow = nit,
                ncol = 3,
                dimnames = list(c(1:nit),
                                c("a^star", "b^star", "c^star")
                                )
                )

bootPred <- matrix(data = NA,
                   nrow = length(nyData$Day),
                   ncol = nit,
                   dimnames = list(c(1:length(nyData$Day)),
                                   c(1:nit))
                   )
```

```r
# Run bootstrap
for (i in 1:nit) {
    # Resample the residuals from fit1
        boot[i,] <- sample(x = fit1$residuals,
                           size = length(fit1$residuals),
                           replace = TRUE
                           )
    # Add those resampled residuals to the original cumulative case data into a
    # dataframe
        temp <- data.frame(cbind(nyData$Day,
                                 boot[i,]+nyData$CaseCumSum)
                           )
    # Run nls() again using the a, b, and c starts from before
        temp <- try(nls(X2 ~ a*exp(b*exp(c*X1)),
                        data = temp,
                        start = list(a = aStart,
                                     b = bStart,
                                     c = cStart)
                        )
        )
    # Save a, b, and c values
        abcEst[i,] <- rbind(temp$m$getPars())
    # Get the predicted Y values for each iteration of the new a, b, and c estimates
        bootPred[,i] <- abcEst[i,1]*exp(abcEst[i,2]*exp(abcEst[i,3]*nyData$Day))
}

# Get the 95% confidence intervals, upper, middle, and lower
nyCIBands2 <- matrix(data = NA,
                     nrow = length(nyData$Day),
                     ncol = 3,
                     dimnames = list(c(1:length(nyData$Day)),
                                     c("lwr", "fit", "upr"))
                     )

for (i in 1:33) {
        nyCIBands2[i,] <- quantile(bootPred[i,], c(0.025, 0.5, 0.975))
}
```

d. Plot the two best-fit curves on top of the plot from (A) and their 95th percentile confidence intervals.



e. Which model fits better and why? (There may be more than one correct answer depending on the metric used; I'm just interested in seeing you justify your decision in a reasonable way.)

I believe that the Gompertz model fits the data better because it captures the shape more accurately (the shallow slope followed by a non-exponential increase and a tapering at the end). Though the glm() produced fit has an incredibly small confidence interval, it's not due to the quality of the fit to the data, but rather due to the fact that that model is the best R can possibly do (as said by Heather in office hours).

e. What is the difference between the interpretation of a confidence interval and that of a prediction interval?

The confidence interval is the range of response values in which we are 95% confident that the true response value lies, while the prediction interval is the range of response values that are expected given a new observation.

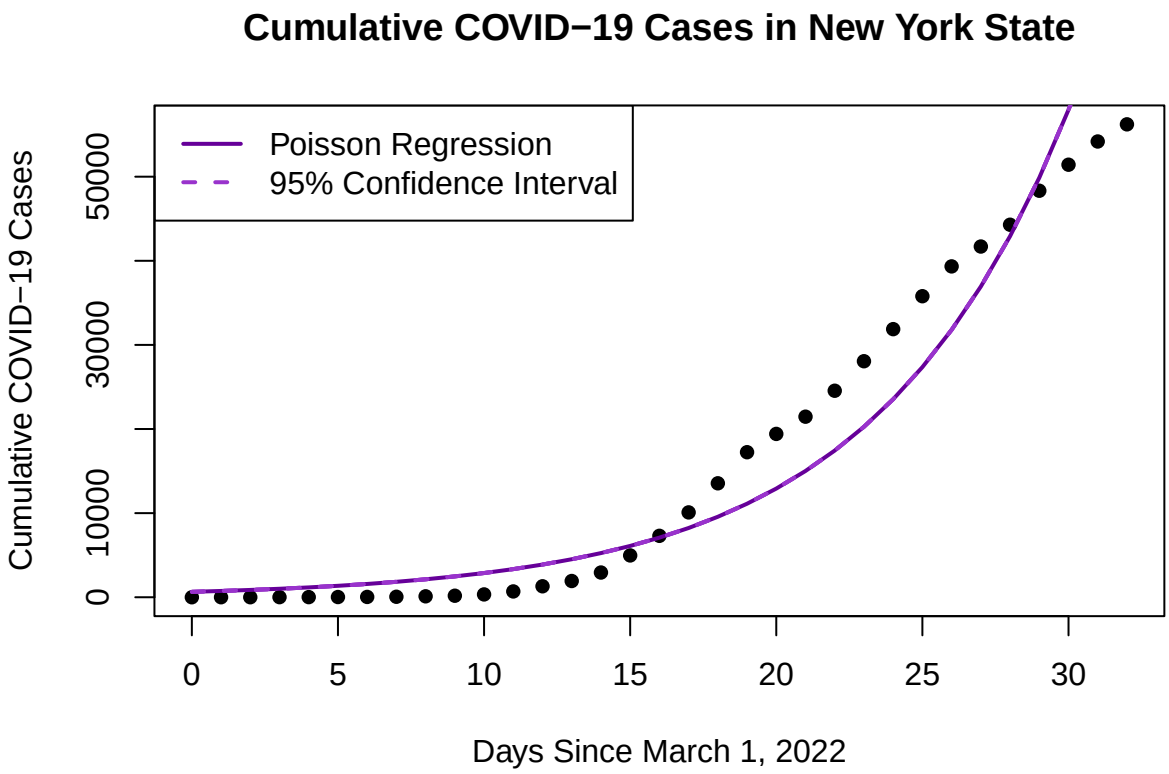
Bonus

Re-do Part B using a Poisson regression. Note that in this case, the Binomial model is more "correct", since there are only so many New Yorkers and each one of them either caught Covid or didn't. (Each person represents a Bernoulli, and collectively, they represent a Binomial.) However, at this point in the epidemic, Covid was still very rare, and so the number of cases is a tiny fraction of the total number of people and so a Poisson model could be used in this case. (This is a situation where we might opt for one model even though we know it can't be correct strictly speaking.) I am curious how the Poisson model fits and so will offer up 5 bonus points for anyone who redoes part B (all parts of B) with a Poisson.

```
# Get Poisson regression, taking inspiration from
# https://www.dataquest.io/blog/tutorial-poisson-regression-in-r/
fit3 <- glm(CaseCumSum ~ Day,
            data = nyData,
            family = poisson(link = "log"))

summary(fit3)
```

```
Call:
glm(formula = CaseCumSum ~ Day, family = poisson(link = "log"),
    data = nyData)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-82.95  -49.85  -34.82   38.26   53.70

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 6.4641037  0.0059307  1089.9   <2e-16 ***
Day         0.1501113  0.0002217   677.1   <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: 784625  on 32  degrees of freedom
Residual deviance:  69472  on 31  degrees of freedom
AIC: 69780

Number of Fisher Scoring iterations: 5
```

```
# Calculate confidence interval bands
nyCIBands3 <- predict.glm(fit3,
                          newData,
                          se.fit = TRUE)
```


Cumulative COVID−19 Cases in New York State
Poisson Regression
95% Confidence Interval
Cumulative COVID−19 Cases
0
10000
30000
50000
0
5
10
15
20
25
30
Days Since March 1, 2022