Homework 4: Diffusion of Tetracycline

Zhang Zile,3230104237

We continue examining the diffusion of tetracycline among doctors in Illinois in the early 1950s, building on our work in lab 6. You will need the data sets `ckm_nodes.csv` and `ckm_network.dat` from the labs.

1. Clean the data to eliminate doctors for whom we have no adoption-date information, as in the labs. Only use this cleaned data in the rest of the assignment.

```
library(tidyverse)
ckm_nodes <- read.csv('data/ckm_nodes.csv',header=T, sep = ",")
noinfor <- which(is.na(ckm_nodes$adoption_date))
ckm_nodes <- ckm_nodes[-noinfor, ]
ckm_network <- read.table('data/ckm_network.dat')
ckm_network <- ckm_network[-noinfor, -noinfor]
```

2. Create a new data frame which records, for every doctor, for every month, whether that doctor began prescribing tetracycline that month, whether they had adopted tetracycline before that month, the number of their contacts who began prescribing strictly *before* that month, and the number of their contacts who began prescribing in that month or earlier. Explain why the dataframe should have 6 columns, and 2125 rows.

```
doctors <- rownames(ckm_nodes)
df <- data.frame(
  doctor = rep(doctors, each = 17),
  month = rep(1:17, times = length(doctors))
)
df <- df |> mutate(that_month = as.numeric(ckm_nodes[doctor,2] == month),
                   that_before = as.numeric(ckm_nodes[doctor,2] < month))
f <- function(x) {
    return(sum(ckm_nodes[ckm_network[as.numeric(x[1]), ] == 1, 2] < as.numeric(x[2])))
}
df <- df |>  mutate(contacts_str_before = apply(df, 1, f))
f <- function(x){
    return(sum(ckm_nodes[ckm_network[as.numeric(x[1]),] == 1, 2] <= as.numeric(x[2])))
}

df <- df  |>  mutate(contacts_in_before = apply(df, 1, f))
```

- There are 6 variables, so the dataframe has 6 columns.
- For every doctor, for every month $\Rightarrow$ 125 doctors $\times$ 17 months $=$ 2125, so the datafram has 2125 rows.

3. Let

$$p_k = \Pr(\text{A doctor starts prescribing tetracycline this month} \mid \text{Number of doctor's contacts prescribing before this month} = k) \tag{1}$$

and

$$q_k = \Pr(\text{A doctor starts prescribing tetracycline this month} \mid \text{Number of doctor's contacts prescribing this month} = k) \tag{2}$$

We suppose that p_k and q_k are the same for all months.

a. Explain why there should be no more than 21 values of k for which we can estimate p_k and q_k directly from the data.

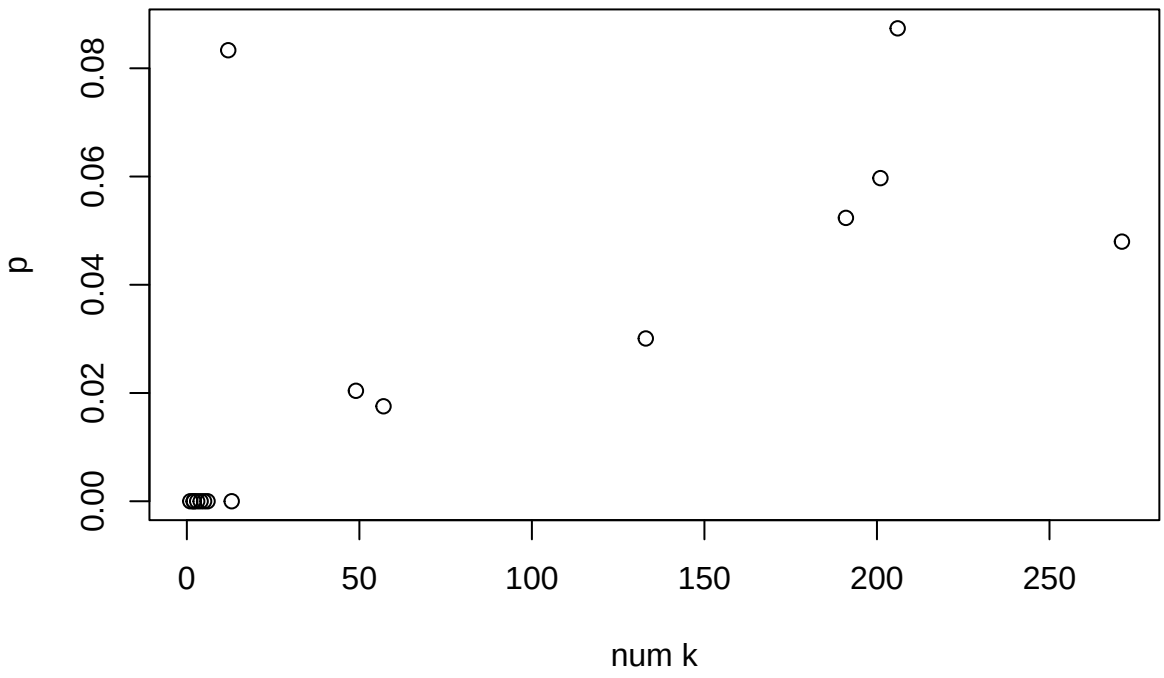
```
max(apply(ckm_network, 1, sum))
```

```
## [1] 20
```

- Because the maximum of the contacts of a doctor is 20, so the possible values of k are from 0 to 20, the number of which is no more than 21.

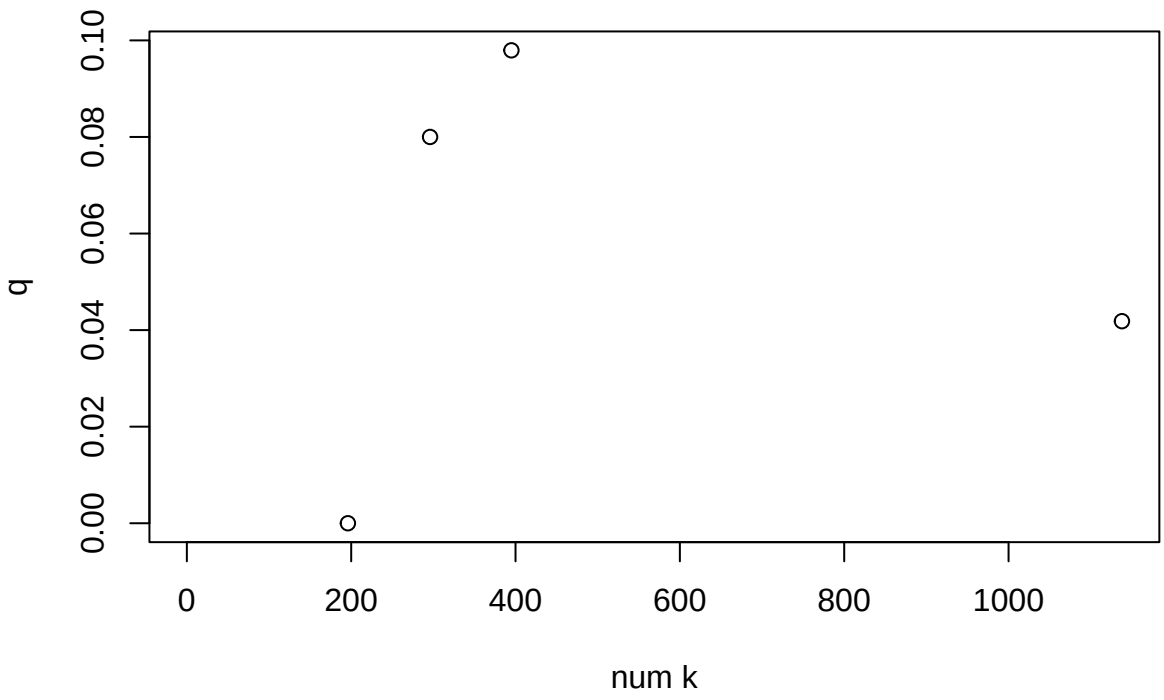
b. Create a vector of estimated p_k probabilities, using the data frame from (2). Plot the probabilities against the number of prior-adoptee contacts k.

```
df_clean <- df[!is.na(df$contacts_str_before), ]
 p.vec <- vector(mode = "numeric",length = 21)
    k.vec <- p.vec
    for(k in 0:20){
        idx <- df_clean$contacts_str_before == k
        k.vec[k+1] <- sum(idx)
         if(is.na(k.vec[k+1]) | k.vec[k+1] == 0) {
        p.vec[k+1] <- NA
    } else {
        dfk <- df_clean[idx, ]
        p1 <- sum(dfk$that_month == 1)
        p.vec[k+1] <- p1 / k.vec[k+1]
    }
    }
    k <- c(0:20)
    plot(k.vec,p.vec, xlab="num k", ylab="p")
```



c. Create a vector of estimated q_k probabilities, using the data frame from (2). Plot the probabilities against the number of prior-or-contemporary-adoptee contacts k.

```
df_clean <- df_clean[!is.na(df_clean$contacts_in_before), ]
p.vec2 <- vector(mode = "numeric",length = 21)
    k.vec2 <- p.vec2
    for(k in 0:20){
        idx <- (df_clean$contacts_in_before - df_clean$contacts_str_before) == k
        k.vec2[k+1] <- sum(idx)
        if(is.na(k.vec2[k+1]) | k.vec2[k+1] == 0){
            p.vec2[k+1] <- NA
        } else{
        dfk <- df_clean[idx,]
        p1 <- sum(dfk$that_month == 1)
        p.vec2[k+1] <- p1/k.vec2[k+1]}
    }
    k <- c(0:20)
    plot(k.vec2 + k.vec, p.vec2, xlab="num k", ylab="q")
```



4. Because it only conditions on information from the previous month, p_k is a little easier to interpret than q_k. It is the probability per month that a doctor adopts tetracycline, if they have exactly k contacts who had already adopted tetracycline.

a. Suppose $p_k = a + bk$. This would mean that each friend who adopts the new drug increases the probability of adoption by an equal amount. Estimate this model by least squares, using the values you constructed in (3b). Report the parameter estimates.

```
df.p <- data.frame(k = 0:20, p = p.vec)
m.1 <- lm(p ~ k, data = df.p)
summary(m.1)
```

```
##
## Call:
## lm(formula = p ~ k, data = df.p)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.029075 -0.015700 -0.001182  0.007297  0.062074
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.056430   0.010564   5.342 0.000104 ***
## k           -0.003908   0.001089  -3.590 0.002956 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02352 on 14 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.4793, Adjusted R-squared:  0.4422
## F-statistic: 12.89 on 1 and 14 DF,  p-value: 0.002956

  Estimate    Std.Error    t   Pr(>|t|)
a  0.056430   0.010564   5.342 0.000104
b -0.003908   0.001089  -3.590 0.002956
```

b. Suppose $p_k = e^{a+bk}/(1+e^{a+bk})$. Explain, in words, what this model would imply about the impact of adding one more adoptee friend on a given doctor's probability of adoption. (You can suppose that $b > 0$, if that makes it easier.) Estimate the model by least squares, using the values you constructed in (3b).

- It is a logistic curve. Suppose $b > 0$. As k grows, the initial stage of growth is approximately exponential (geometric); then, as saturation begins, the growth slows to linear (arithmetic), and at maturity, growth stops.

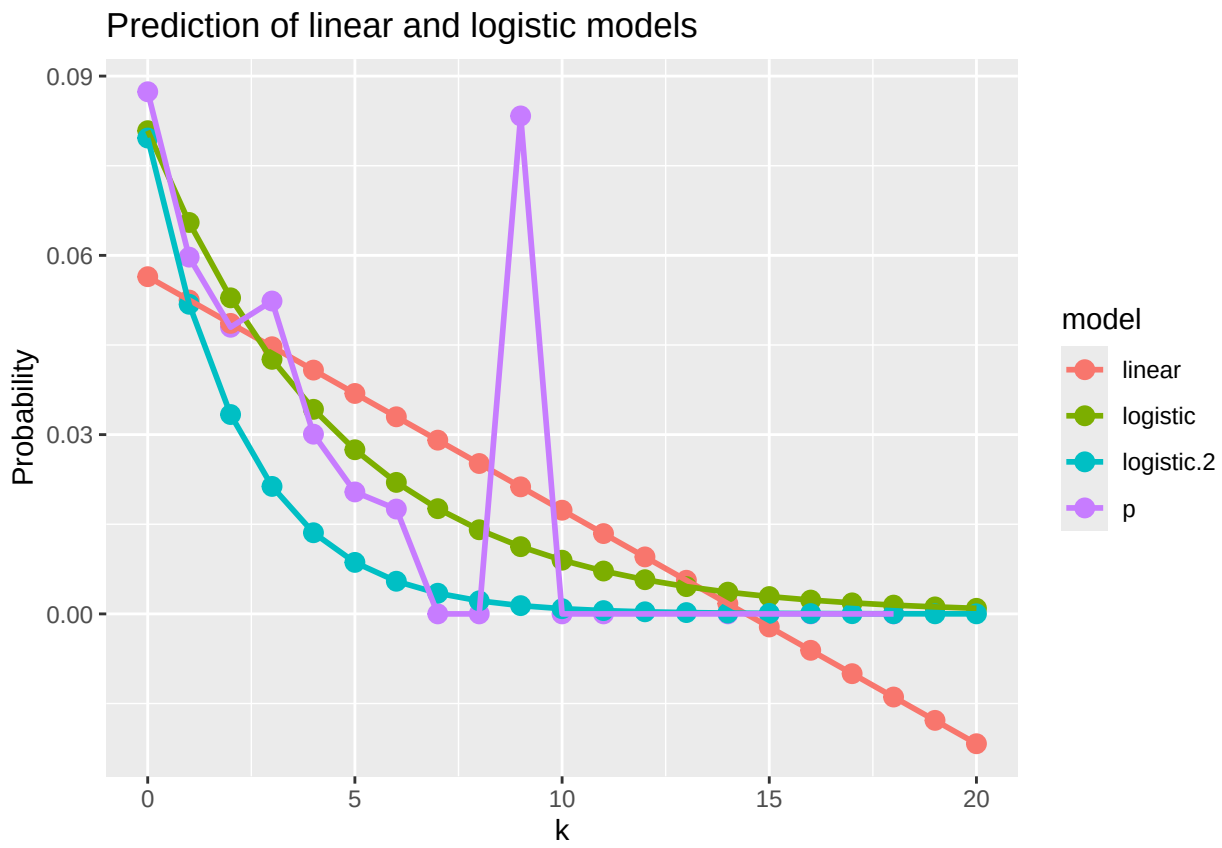
```
f <- function(k,a,b){
        return(exp(a+b*k)/(1+exp(a+b*k)))
    }
    m.2 <- nls(p ~ f(k, a, b), data = df.p, start = list(a = 0, b = -0.2))
    summary(m.2)
```

```
##
## Formula: p ~ f(k, a, b)
##
## Parameters:
##   Estimate Std. Error t value Pr(>|t|)
## a -2.43067    0.21457 -11.328 1.95e-08 ***
## b -0.22725    0.07407  -3.068  0.00834 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02096 on 14 degrees of freedom
##
## Number of iterations to convergence: 8
## Achieved convergence tolerance: 2.37e-06
##   (5 observations deleted due to missingness)
```

c. Plot the values from (3b) along with the estimated curves from (4a) and (4b). (You should have one plot, with k on the horizontal axis, and probabilities on the vertical axis .) Which model do you prefer, and why?

```
m.3 <- lm(p.log ~ k, df.p  |>
                  mutate(p.log = ifelse(p==0, log(0.0001/(1-0.0001)), log(p/(1-p)))))
m1 = predict(m.1, newdata=data.frame(k=c(0:20)))
    m2 = predict(m.2, newdata=data.frame(k=c(0:20)))
    df.p <- mutate(df.p, linear = m1, logistic = m2)
    t = predict(m.3, newdata=data.frame(k=c(0:20)))
```

```
m3 = exp(t)/(1+exp(t))
df.p <- mutate(df.p, linear = m1, logistic = m2, logistic.2 = m3)
df.tidy <- df.p  |>  gather(model, res, -k)  |>  na.omit()
df.tidy  |>  ggplot() + geom_point(aes(x = k, y = res, color = model), size = 3) +
      geom_line(aes(x = k, y = res, color = model), linewidth = 1) +
      labs(y = "Probability",
           title = "Prediction of linear and logistic models")
```



* The linear model is the worst. I use two methods to establish the logistic model and I think green line `logistic` using function `nls()` looks better on the whole. Because `logistic.2` seems a little overfitting in the tail part.

For quibblers, pedants, and idle hands itching for work to do: The p_k values from problem 3 aren't all equally precise, because they come from different numbers of observations. Also, if each doctor with k adoptee contacts is independently deciding whether or not to adopt with probability p_k, then the variance in the number of adoptees will depend on p_k. Say that the actual proportion who decide to adopt is $\hat{p}_k$. A little probability (exercise!) shows that in this situation, $\mathbb{E}[\hat{p}_k] = p_k$, but that $\mathrm{Var}[\hat{p}_k] = p_k(1-p_k)/n_k$, where n_k is the number of doctors in that situation. (We estimate probabilities more precisely when they're really extreme [close to 0 or 1], and/or we have lots of observations.) We can estimate that variance as $\hat{V}_k = \hat{p}_k(1-\hat{p}_k)/n_k$. Find the $\hat{V}_k$, and then re-do the estimation in (4a) and (4b) where the squared error for p_k is divided by $\hat{V}_k$. How much do the parameter estimates change? How much do the plotted curves in (4c) change?

```
idx <- !(is.na(p.vec) | p.vec == 0)
wt <- p.vec*(1-p.vec)/k.vec
m.w1 <- lm(p ~ k, data = df.p[idx,], weight = 1/wt[idx])
summary(m.w1)
```

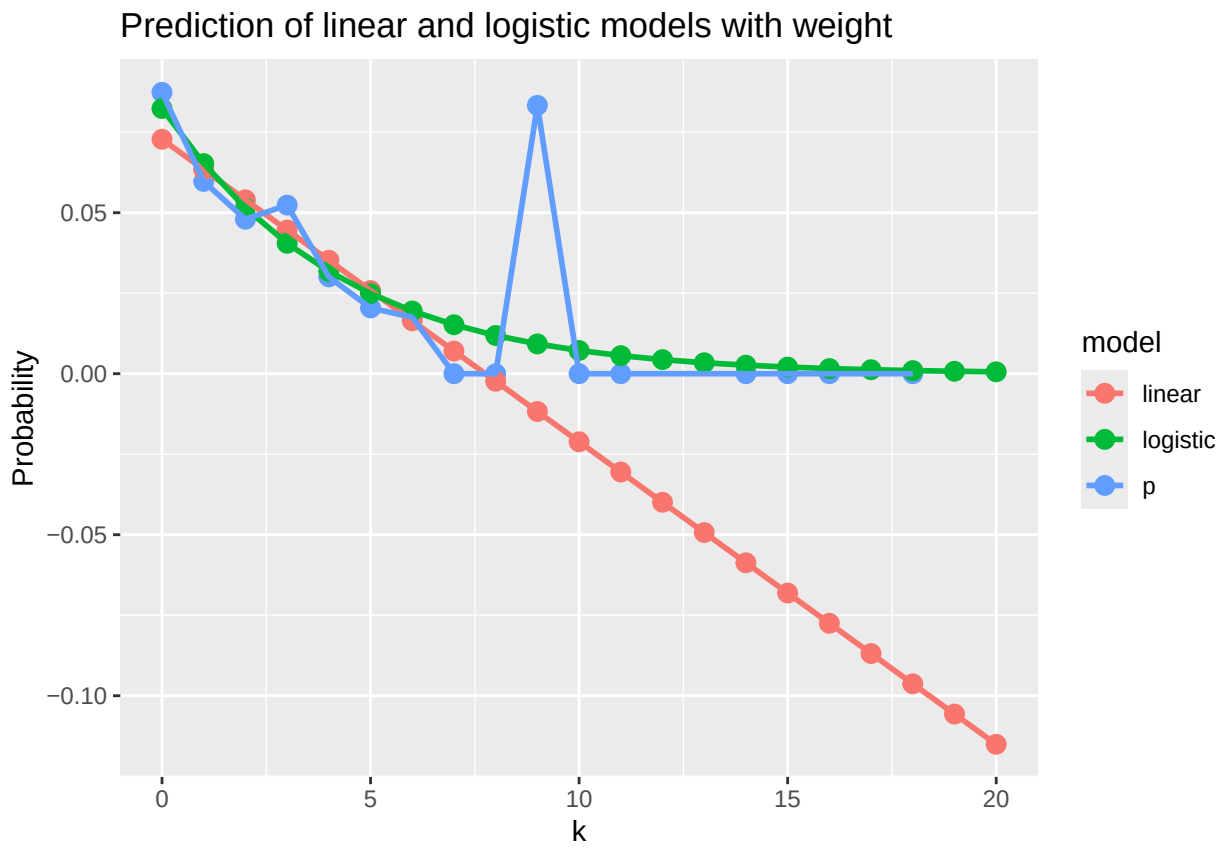
```
##
## Call:
## lm(formula = p ~ k, data = df.p[idx, ], weights = 1/wt[idx])
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46593 -0.28854 -0.07934  0.54492  1.19164
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.072808   0.007684   9.475 7.87e-05 ***
## k           -0.009394   0.002181  -4.307  0.00505 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6661 on 6 degrees of freedom
## Multiple R-squared:  0.7556, Adjusted R-squared:  0.7149
## F-statistic: 18.55 on 1 and 6 DF,  p-value: 0.005052
```

```
m.w2 <- nls(p ~ f(k, a, b), data = df.p[idx,],
            start = list(a = 0, b = -0.2), weight = 1/wt[idx])
summary(m.w2)
```

```
##
## Formula: p ~ f(k, a, b)
##
## Parameters:
##   Estimate Std. Error t value Pr(>|t|)
## a -2.41074    0.10809 -22.303 5.31e-07 ***
## b -0.25140    0.04862  -5.171  0.00207 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5364 on 6 degrees of freedom
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 9.083e-07
```

```
w1 = predict(m.w1, newdata=data.frame(k=c(0:20)))
w2 = predict(m.w2, newdata=data.frame(k=c(0:20)))

df.w <- data.frame(k = 0:20, p = p.vec)
df.w <- mutate(df.w, linear = w1, logistic = w2)
dft <- df.w  |>  gather(model, res, -k)  |>  na.omit()
dft |>  ggplot() + geom_point(aes(x = k, y = res, color = model), size = 3) +
      geom_line(aes(x = k, y = res, color = model), linewidth = 1) +
      labs(y = "Probability",
           title = "Prediction of linear and logistic models with weight")
```



** The change of plots**: Both curves are more fitting when k is small and seem to somehow neglect the dots in the middle-top. From my analysis, the reason is that smaller samples cause larger estimated variances, which reduce the weight of these points. And there are more observations when k is small, which adds to the weight.