Homework 4: Diffusion of Tetracycline

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')
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]
nrow(ckm_nodes)
```

```
## [1] 125
```

```
dim(ckm_network)
```

```
## [1] 125 125
```

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.

```
library(dplyr)
library(purrr)

max_months <- 17
doctor_month_list <- lapply(1:nrow(ckm_nodes), function(i) {
  adoption_month <- min(ckm_nodes$adoption_date[i], max_months)
  months <- 1:adoption_month
  neighbors <- which(ckm_network[i, ] == 1)

  data.frame(
    doctor_id = i,
    month = months,
    began_this_month = ifelse(months == adoption_month, 1, 0),
    adopted_before_this_month = ifelse(months < adoption_month, 0, NA_real_),
    count_prior_adopters = vapply(months, function(t)
      sum(ckm_nodes$adoption_date[neighbors] < t, na.rm = TRUE), numeric(1)),
    count_prior_contemporary_adopters = vapply(months, function(t)
      sum(ckm_nodes$adoption_date[neighbors] <= t, na.rm = TRUE), numeric(1))
  )
})

doctor_month_df <- bind_rows(doctor_month_list)
cat("Number of rows in the doctor-month data frame:", nrow(doctor_month_df), "\n")
```

```
## Number of rows in the doctor-month data frame: 981
```

```
cat("Number of columns in the doctor-month data frame:", ncol(doctor_month_df), "\n")
```

```
## Number of columns in the doctor-month data frame: 6
```

```
head(doctor_month_df)
```

```
##   doctor_id month began_this_month adopted_before_this_month
## 1         1     1                1                        NA
## 2         2     1                0                         0
## 3         2     2                0                         0
## 4         2     3                0                         0
## 5         2     4                0                         0
## 6         2     5                0                         0
##   count_prior_adopters count_prior_contemporary_adopters
## 1                    0                                 1
## 2                    0                                 0
## 3                    0                                 0
## 4                    0                                 1
## 5                    1                                 2
## 6                    2                                 2
```

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_degree <- max(rowSums(ckm_network))
max_degree
```

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

The maximum number of k values for direct estimation is 21 because: The data contains nodes with degrees from 0 to 20 (21 values). Degrees larger than 20 are unobserved, making estimation impossible without additional assumptions or extrapolation.

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.

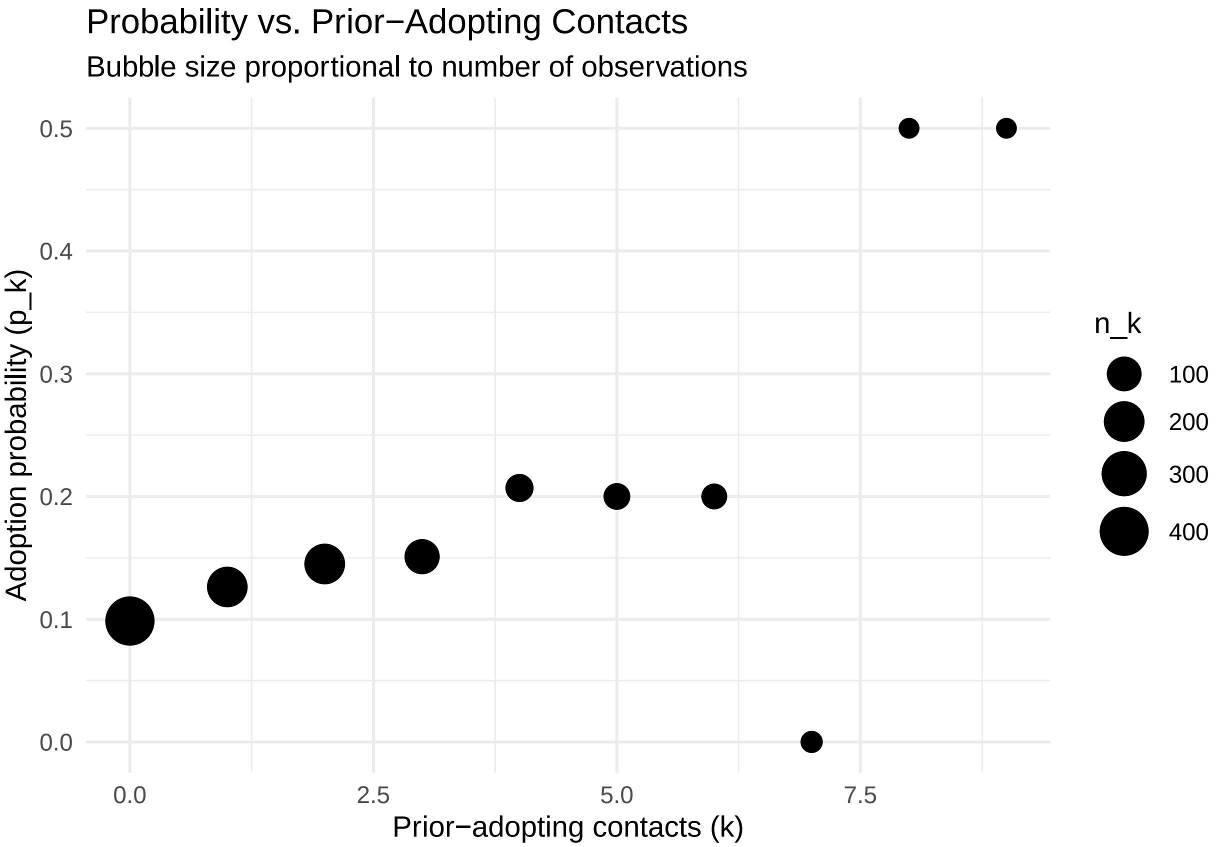
```
p_k_df <- doctor_month_df %>%
  group_by(k = count_prior_adopters) %>%
  summarise(p_k = mean(began_this_month), n_k = n()) %>%
  filter(n_k > 0)

cat("k value range:", min(p_k_df$k), "-", max(p_k_df$k), "\n")
```

```
## k value range: 0 - 9
```

```
cat("number of estimable k values:", nrow(p_k_df), "\n")
```

```
## number of estimable k values: 10
```

```
ggplot(p_k_df, aes(x = k, y = p_k)) +
  geom_point(aes(size = n_k)) +
  scale_size_continuous(range = c(3, 8)) +
  labs(x = "Prior-adopting contacts (k)", y = "Adoption probability (p_k)",
       title = "Probability vs. Prior-Adopting Contacts",
       subtitle = "Bubble size proportional to number of observations") +
  theme_minimal()
```



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.

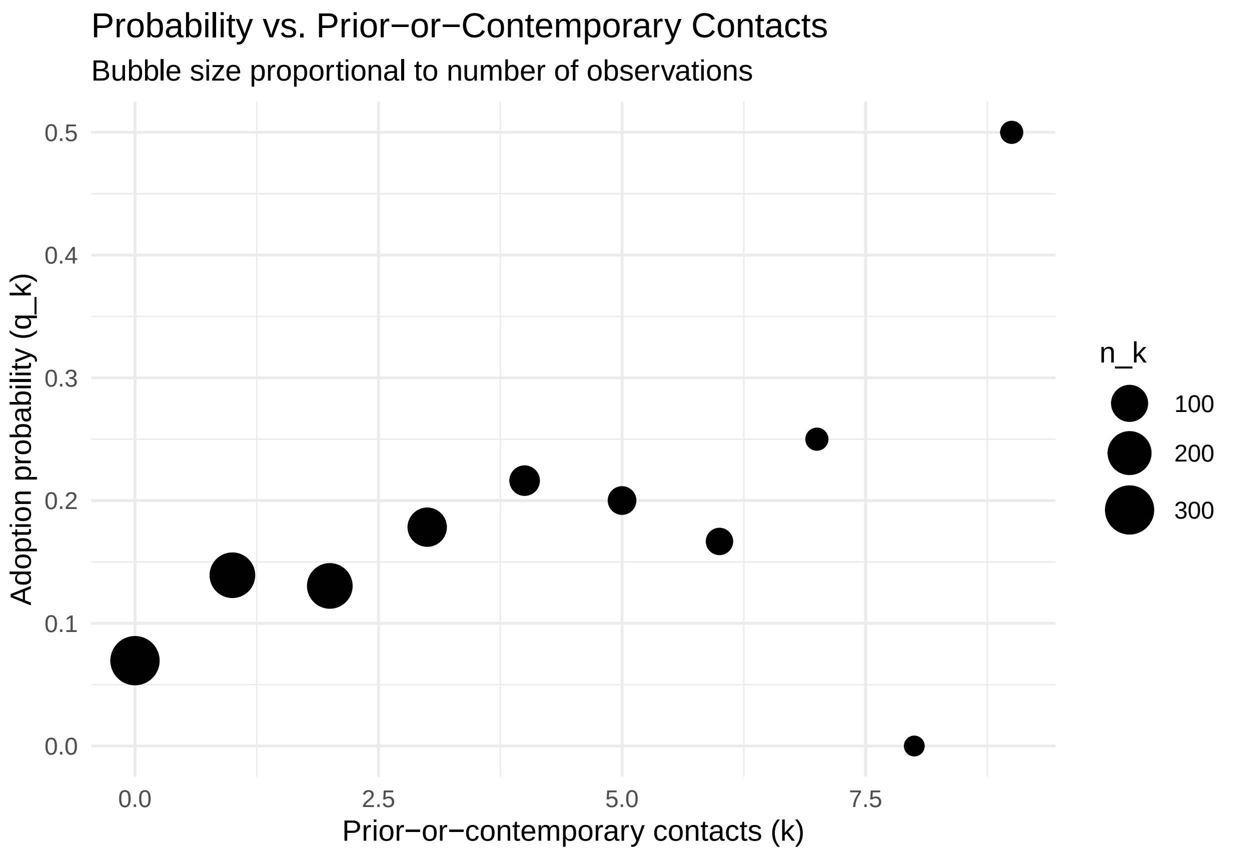
```
q_k_df <- doctor_month_df %>%
  group_by(k = count_prior_contemporary_adopters) %>%
  summarise(q_k = mean(began_this_month), n_k = n()) %>%
  filter(n_k > 0)

cat("k value range:", min(q_k_df$k), "-", max(q_k_df$k), "\n")
```

```
## k value range: 0 - 9
```

```
cat("number of estimable k values:", nrow(q_k_df), "\n")
```

```
## number of estimable k values: 10
```

```r
ggplot(q_k_df, aes(x = k, y = q_k)) +
  geom_point(aes(size = n_k)) +
  scale_size_continuous(range = c(3, 8)) +
  labs(x = "Prior-or-contemporary contacts (k)", y = "Adoption probability (q_k)",
       title = "Probability vs. Prior-or-Contemporary Contacts",
       subtitle = "Bubble size proportional to number of observations") +
  theme_minimal()
```



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.

```r
linear_model <- lm(p_k ~ k, data = p_k_df)
linear_coef <- coef(linear_model)
cat("Linear model coefficients:\n")
```

```
## Linear model coefficients:
```

```r
cat("Intercept (a):", linear_coef[1], "\n")
```

```
## Intercept (a): 0.05881515
```

```r
cat("Slope (b):", linear_coef[2], "\n")
```

```
## Slope (b): 0.03421052
```

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).

```
logistic_model <- nls(
  p_k ~ exp(a + b*k)/(1 + exp(a + b*k)),
  data = p_k_df,
  start = list(a = 0, b = 0)
)
logistic_coef <- coef(logistic_model)
cat("\nLogistic model coefficients:\n")
```

```
##
## Logistic model coefficients:
```

```
cat("a:", logistic_coef[1], "\n")
```

```
## a: -2.690262
```

```
cat("b:", logistic_coef[2], "\n")
```

```
## b: 0.2618798
```

```
cat("\nInterpretation: Each additional adopting neighbor")
```

```
##
## Interpretation: Each additional adopting neighbor
```

```
cat("\nincreases the log-odds of adoption by", round(logistic_coef[2], 3))
```

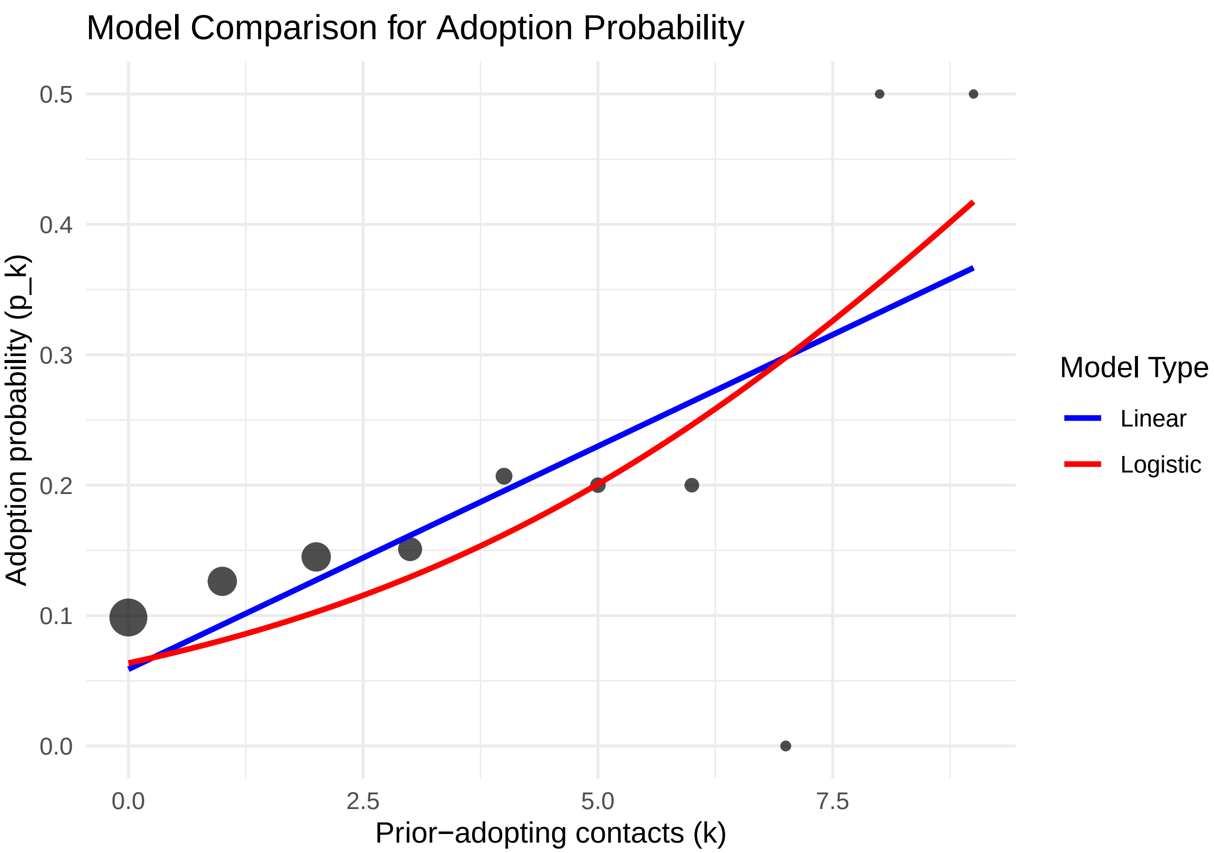
```
##
## increases the log-odds of adoption by 0.262
```

```
cat("\nwhich corresponds to multiplying the odds by", round(exp(logistic_coef[2]), 3))
```

```
##
## which corresponds to multiplying the odds by 1.299
```

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?

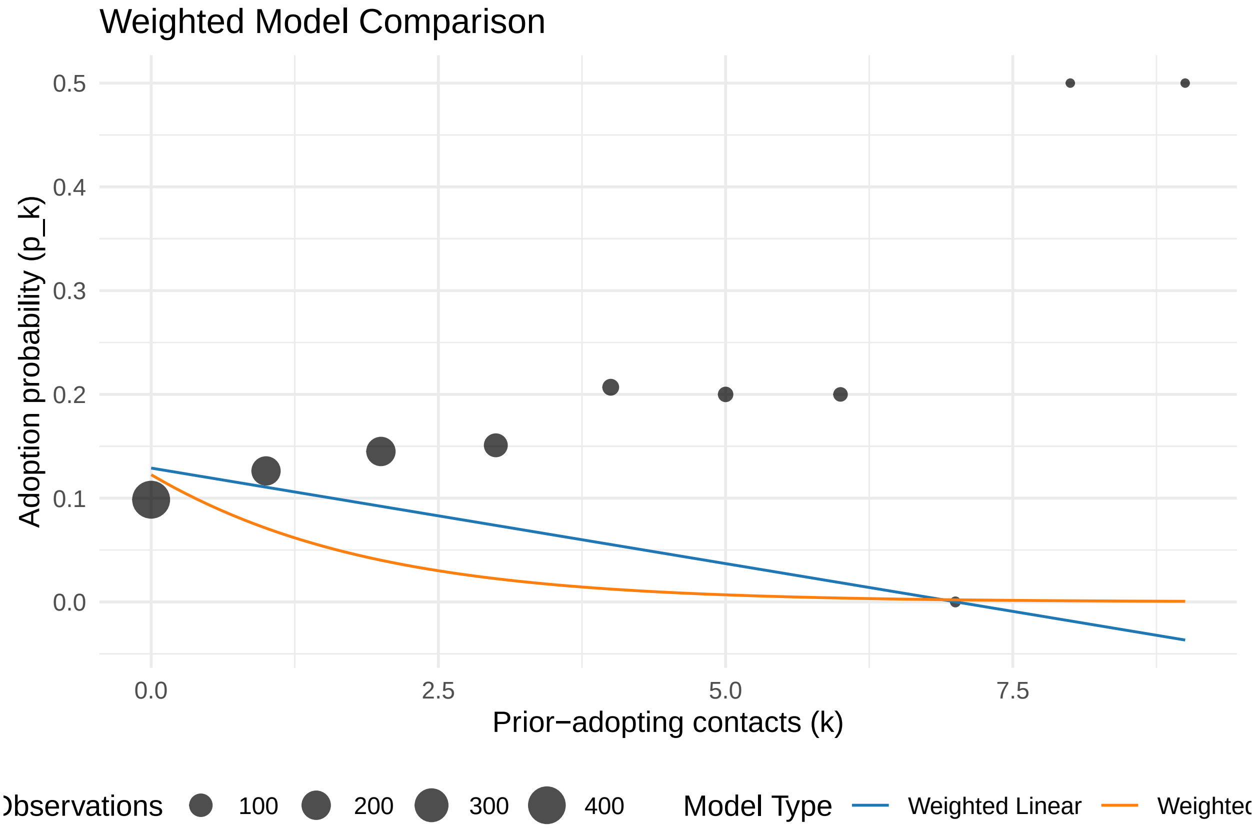
```
k_range <- data.frame(k = seq(0, max(p_k_df$k), by = 0.1))
k_range$linear <- predict(linear_model, newdata = k_range)
k_range$logistic <- predict(logistic_model, newdata = k_range)

ggplot(p_k_df, aes(x = k, y = p_k)) +
  geom_point(aes(size = n_k), alpha = 0.7) +
  geom_line(data = k_range, aes(y = linear, color = "Linear"), linewidth = 1) +
  geom_line(data = k_range, aes(y = logistic, color = "Logistic"), linewidth = 1) +
  scale_color_manual(values = c(Linear = "blue", Logistic = "red")) +
  labs(x = "Prior-adopting contacts (k)", y = "Adoption probability (p_k)",
       title = "Model Comparison for Adoption Probability",
       color = "Model Type") +
  theme_minimal() +
  guides(size = "none")
```



The logistic model provides a more robust and theoretically sound representation of how social contacts influence adoption behavior. It quantifies both the baseline propensity and the saturating effect of social networks, making it a superior choice for modeling the probability of doctors adopting tetracycline based on their peers' decisions.

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 $\text{Var}[\hat{p}_k] = p_k(1 - p_k)/n_k$, 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?

```
p_k_df <- p_k_df %>%
  mutate(
    variance = p_k * (1 - p_k) / n_k,
    variance = ifelse(variance <= 0 | is.na(variance), 1e-6, variance)
  )

weighted_linear <- lm(p_k ~ k, data = p_k_df, weights = 1/variance)

weighted_logistic <- nls(
  p_k ~ exp(a + b*k)/(1 + exp(a + b*k)),
  data = p_k_df,
  weights = 1/variance,
  start = list(a = 0, b = 0),
  control = nls.control(warnOnly = TRUE)
)

k_range$weighted_linear <- predict(weighted_linear, newdata = k_range)
k_range$weighted_logistic <- predict(weighted_logistic, newdata = k_range)
ggplot(p_k_df, aes(x = k, y = p_k)) +
  geom_point(aes(size = n_k), alpha = 0.7) +
  geom_line(data = k_range, aes(y = weighted_linear, color = "Weighted Linear")) +
  geom_line(data = k_range, aes(y = weighted_logistic, color = "Weighted Logistic")) +
  scale_color_manual(values = c("Weighted Linear" = "#1f77b4", "Weighted Logistic" = "#ff7f0e")) +
  labs(x = "Prior-adopting contacts (k)", y = "Adoption probability (p_k)",
       title = "Weighted Model Comparison",
       color = "Model Type", size = "Observations") +
  theme_minimal() +
  theme(legend.position = "bottom")
```


Weighted Model Comparison
Adoption probability (p_k)
Prior−adopting contacts (k)
)bservations
Model Type
Weighted Linear
Weightec