Problem Set 1 Solution 2021

```
## Warning: package 'tidyverse' was built under R version 3.6.3
## Warning: package 'ggplot2' was built under R version 3.6.2
## Warning: package 'tibble' was built under R version 3.6.3
## Warning: package 'tidyr' was built under R version 3.6.3
## Warning: package 'purrr' was built under R version 3.6.3
## Warning: package 'dplyr' was built under R version 3.6.3
## Warning: package 'forcats' was built under R version 3.6.1
## Warning: package 'lmtest' was built under R version 3.6.2
```

I. Rudimentary understanding of logistic regression analysis

Upon mastery of this problem, a student should be able to:

- teach logistic regression for binary responses to a health scientist or professional; interpret regression coefficients as log odds ratios
- graphically display binary data and make sensible guestimates of the coefficients that would be obtained from a simple logistic regression (SLR)

Simulate a data set of 250 observations that satisfy each of logistic models A and B below in which $\log \text{odds}(Y = 1) = \beta_0 + \beta_1 X$, for $X_i $ iid uniform(0,1):

$$\text{Model A} : \beta_0 = -2, \beta_1 = 2 \text{Model B} : \beta_0 = -2, \beta_1 = 4$$

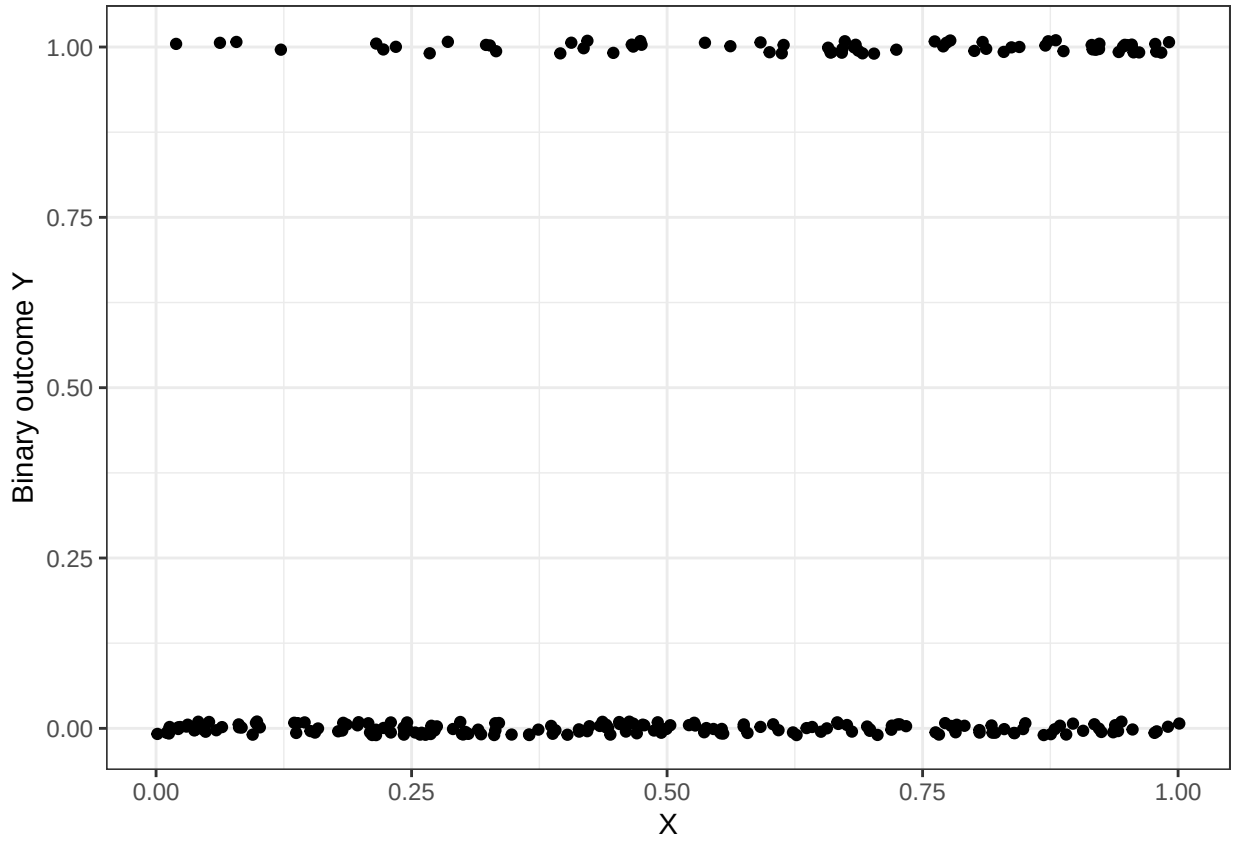
NOTE: Your solution will likely differ from this solution given the random seed should vary from student to student (or workgroup to workgroup).

For each data set, fill in the table below by completing the following:

1. plot Y against X, jittering the Ys so you can see all of the observations

```
set.seed(101)
X = runif(n = 250)
logOR_modelA = -2 + 2 * X
logOR_modelB = -2 + 4 * X
#We use the "expit" or "inverse logit function" to convert log odds to probabilities.
expit = function(x) {
  p = exp(x)/(1 + exp(x))
  return(p)
}
prob_modelA = expit(logOR_modelA)
prob_modelB = expit(logOR_modelB)
# Generate y, size = 1 because we only have one trial for each observation
Y_modelA = rbinom(n = 250, size = 1, prob = prob_modelA)
```

```r
Y_modelB = rbinom(n = 250, size = 1, prob = prob_modelB)
df = data.frame(model = rep(c("A", "B"), each = 250), X = rep(X, 2), Y = c(Y_modelA, Y_modelB))
df %>% filter(model == "A") %>%
  ggplot(aes(x = X, y = Y)) + theme_bw() +
  labs(x = "X", y = "Binary outcome Y") +
  geom_point(position = position_jitter(h = 0.01, w = 0.01))
```



2. Split the data set into 5 roughly equal-sized X strata by using cut points: 0.2, 0.4, 0.6, 0.8

```r
df$Xgrp = ceiling(df$X * 5)
tab = df %>%
    group_by(model, Xgrp) %>%
    summarise(trials = n(), succ = sum(Y), p = succ / trials, odds = p / (1 - p), lodds = log(odds))
```

```
## `summarise()` regrouping output by 'model' (override with `.groups` argument)
```

```r
tab
```

```
## # A tibble: 10 x 7
## # Groups:   model [2]
##    model  Xgrp trials  succ      p  odds   lodds
##    <fct> <dbl>  <int> <int>  <dbl> <dbl>   <dbl>
##  1 A         1     42     4 0.0952 0.105 -2.25
##  2 A         2     55     9 0.164  0.196 -1.63
##  3 A         3     50    12 0.24   0.316 -1.15
##  4 A         4     47    18 0.383  0.621 -0.477
```

```
##  5 A         5     56    27 0.482  0.931 -0.0715
##  6 B         1     42     4 0.0952 0.105 -2.25
##  7 B         2     55    15 0.273  0.375 -0.981
##  8 B         3     50    22 0.44   0.786 -0.241
##  9 B         4     47    36 0.766  3.27   1.19
## 10 B         5     56    41 0.732  2.73   1.01
```

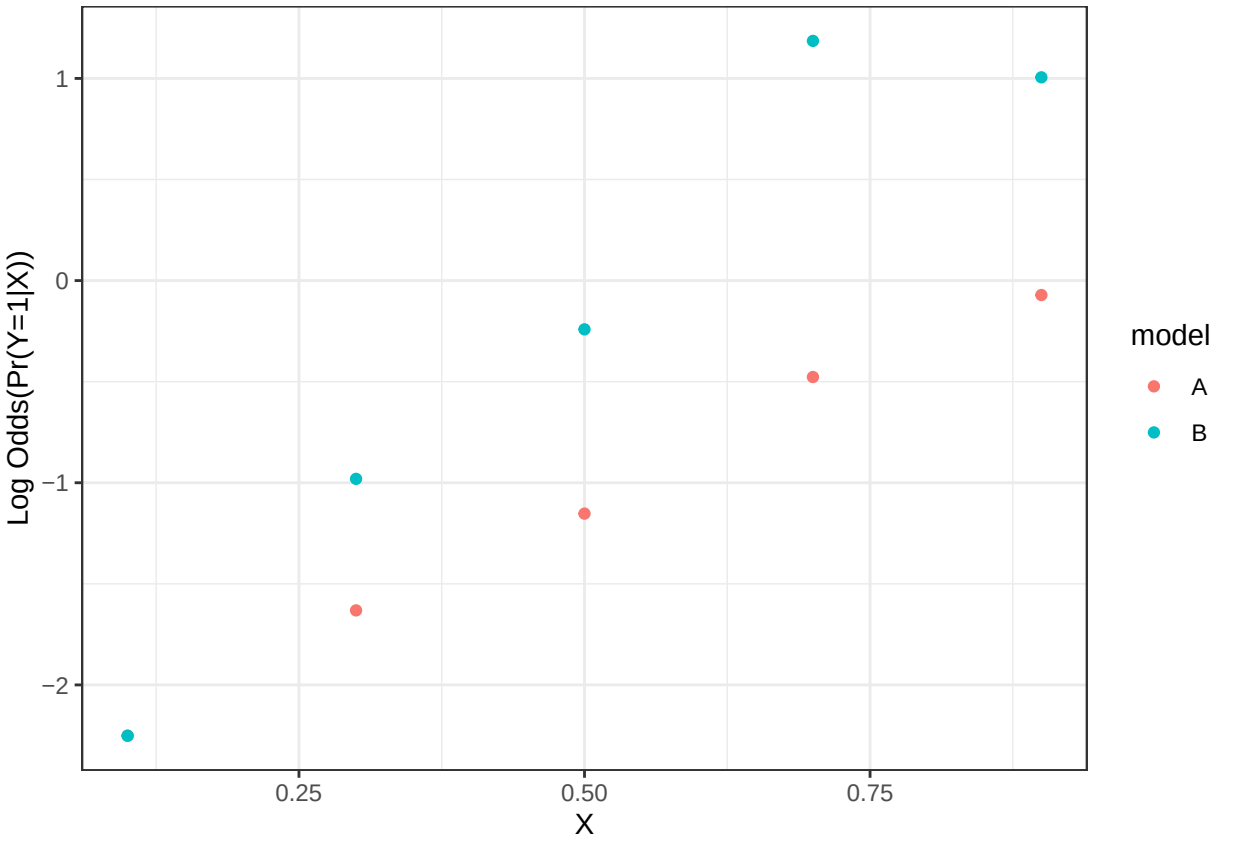
3. Estimate the log odds(Y=1) in each stratum in the table provided below

```
tab$lodds
```

```
##  [1] -2.25129180 -1.63141682 -1.15267951 -0.47692407 -0.07145896 -2.25129180
##  [7] -0.98082925 -0.24116206  1.18562367  1.00552187
```

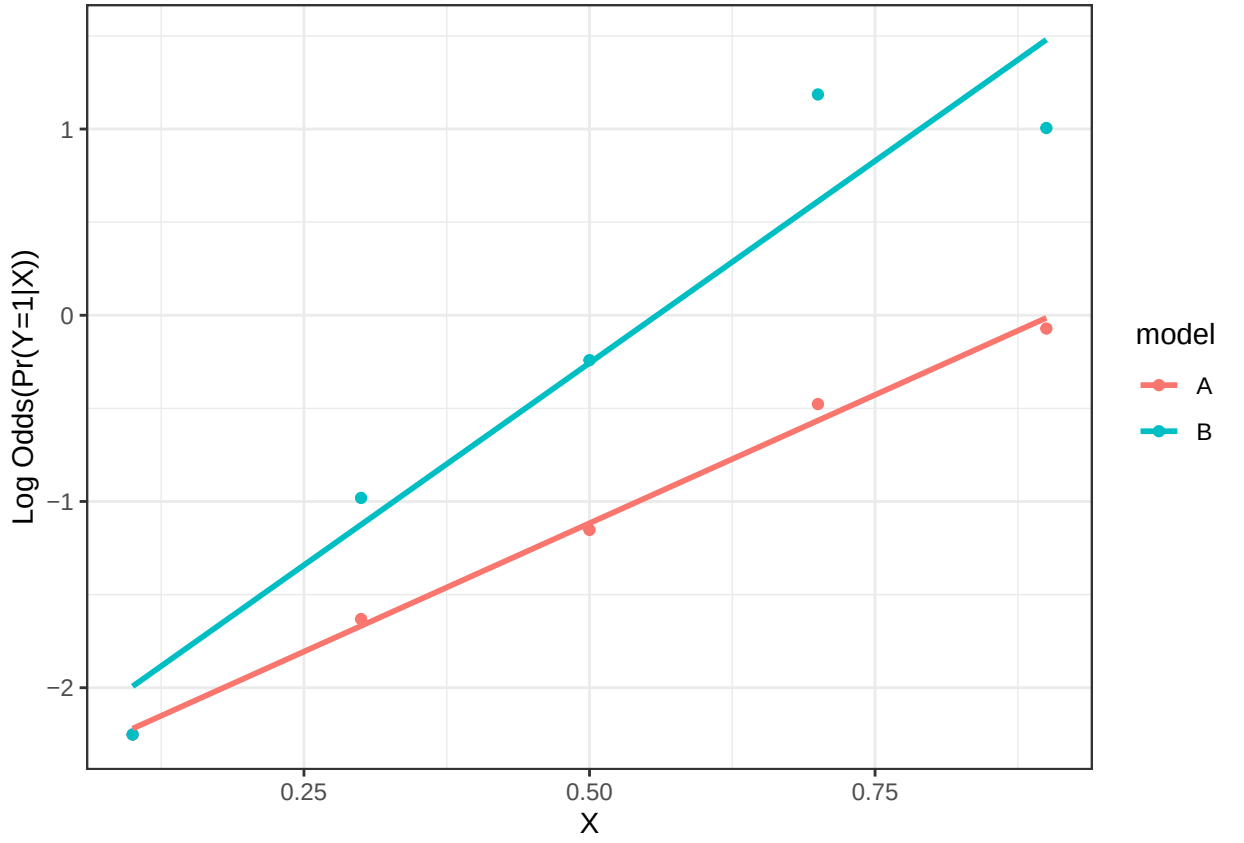
4. Plot the log odds(Y=1) against the mid-point of each X stratum

```
tab$Xgrp_midpoint = rep(c(0.1, 0.3, 0.5, 0.7, 0.9), 2)
ggplot(tab, aes(x = Xgrp_midpoint, y = lodds, color = model)) + theme_bw() +
  geom_point() +
  labs(x = "X", y = "Log Odds(Pr(Y=1|X))") +
  theme_bw()
```



5. Estimate the intercept and slope for the plot of log odds(Y=1) against stratum midpoints using the graph.

```r
ggplot(tab, aes(x = Xgrp_midpoint, y = lodds, color = model)) +
  geom_point() + theme_bw() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "X", y = "Log Odds(Pr(Y=1|X))") +
  theme_bw()
```



6. Determine the predicted log odds at the midpoint of each bin from the fitted line

```r
# fit a linear model with a different intercept and slope for models A and B
fit_A = lm(lodds ~ Xgrp_midpoint , data = tab[tab$model=="A",])
summary(fit_A)
```

```
##
## Call:
## lm(formula = lodds ~ Xgrp_midpoint, data = tab[tab$model == "A",
##     ])
##
## Residuals:
##        1        2        3        4        5
## -0.03171  0.03675 -0.03593  0.08841 -0.05754
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.49529    0.06374  -39.15 3.67e-05 ***
## Xgrp_midpoint  2.75708    0.11096   24.85 0.000143 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07018 on 3 degrees of freedom
## Multiple R-squared:  0.9952, Adjusted R-squared:  0.9936
## F-statistic: 617.4 on 1 and 3 DF,  p-value: 0.0001429
```

```
fit_B = lm(lodds ~ Xgrp_midpoint , data = tab[tab$model=="B",])
summary(fit_B)
```

```
##
## Call:
## lm(formula = lodds ~ Xgrp_midpoint, data = tab[tab$model == "B",
##     ])
##
## Residuals:
##        1        2        3        4        5
## -0.25885  0.14361  0.01527  0.57404 -0.47407
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -2.4264     0.4202  -5.774  0.01033 *
## Xgrp_midpoint   4.3400     0.7315   5.933  0.00957 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4626 on 3 degrees of freedom
## Multiple R-squared:  0.9215, Adjusted R-squared:  0.8953
## F-statistic:  35.2 on 1 and 3 DF,  p-value: 0.00957
```

7. Calculate the corresponding predicted probability that Y=1 given X as a function of X.

```
tab$lodds_pred = c(fitted.values(fit_A),fitted.values(fit_B))
tab$p_pred = expit(tab$lodds_pred)
tab
```

```
## # A tibble: 10 x 10
## # Groups:   model [2]
##    model  Xgrp trials  succ      p  odds   lodds Xgrp_midpoint lodds_pred p_pred
##    <fct> <dbl>  <int> <int>  <dbl> <dbl>   <dbl>         <dbl>      <dbl>  <dbl>
##  1 A         1     42     4 0.0952 0.105 -2.25            0.1    -2.22   0.0980
##  2 A         2     55     9 0.164  0.196 -1.63            0.3    -1.67   0.159
##  3 A         3     50    12 0.24   0.316 -1.15            0.5    -1.12   0.247
##  4 A         4     47    18 0.383  0.621 -0.477           0.7    -0.565  0.362
##  5 A         5     56    27 0.482  0.931 -0.0715          0.9    -0.0139 0.497
##  6 B         1     42     4 0.0952 0.105 -2.25            0.1    -1.99   0.120
##  7 B         2     55    15 0.273  0.375 -0.981           0.3    -1.12   0.245
##  8 B         3     50    22 0.44   0.786 -0.241           0.5    -0.256  0.436
##  9 B         4     47    36 0.766  3.27   1.19            0.7     0.612  0.648
## 10 B         5     56    41 0.732  2.73   1.01            0.9     1.48   0.815
```

8. You have successfully conducted two simple logistic regressions by hand. Now write a short paragraph that explains logistic regression to a layperson in your own words.

Standard linear regression analysis is not appropriate when we have a binary response variable (Y), i.e. a two- level categorical variable, for example: death (1 for dead, 0 for survived) or relapse (1 for relapse, 0

for remission). If our response variable is binary, the mean or expected value of the binary response is the probability of a "positive" outcome given a set of predictor variables X, i.e. Pr(Y = 1|X). We know that a probability is at least 0 and at most 1; however, using linear regression analysis to model Pr(Y=1|X) can yield predicted probabilities that are less than 0 or greater than 1. Logistic regression analysis allows us to model Pr(Y = 1|X) in such a way that our predicted probabilities are within the required bounds for all X. Specifically, we transform the probability of a "positive" outcome, Pr(Y = 1|X), into the log odds of a "positive" outcome, log(Pr(Y = 1|X)/Pr(Y = 0|X)) and then perform linear regression on the transformed variable. After fitting an appropriate model, the results may be transformed back to the probability scale.

II. Connection of logistic regression to 2x2 tables; confounding and effect modification

Upon mastery of this problem, a student should be able to:

- Create one or multiple 2x2 tables from which to estimate log odds ratios that correspond to coefficients from simple or multiple logistic regressions
- Appreciate the invariance of the odds ratio as one important reason logistic regression is popular in epidemiology
- Pool log odds ratios across strata using weighted averages as an approximation to logistic regression

Use the National Medical Expenditure Survey (NMES) data set for this problem. The general goal is to describe the association of self-reported smoking with the indicator of major smoking-caused disease (mscd), a group of diseases the U.S. Surgeon General and WHO say are caused by smoking.

Part A: Simple logistic regression

1. Define a variable mscd to represent whether or not a person has a major smoking caused disease (e.g. lc5 or chd5 =1). Make a 2x2 table of mscd against eversmk (1-yes; 0-no). Calculate the log odds ratio, its standard error and 95% CI using 652 methods for 2x2 tables. To simplify the analysis, drop those people who have a missing value of eversmk (this is to simplify the exercise and is not generally an acceptable strategy).

```
load('./nmes.rdata')
nmes <- nmes[!(nmes$eversmk=="."), ] %>%
  mutate(mscd = ifelse((lc5 | chd5), 1, 0),
         eversmk = as.numeric(eversmk))
tab <- table(nmes$mscd, nmes$eversmk)
tab
```

```
##
##        0    1
##   0 4626 5739
##   1  433  886
```

```
logOR = log((tab[1, 1] * tab[2, 2])/(tab[1, 2] * tab[2, 1]))
logOR
```

```
## [1] 0.5003868
```

```
SE_logOR = sqrt(1/tab[1, 1] + 1/tab[1, 2] + 1/tab[2, 1] + 1/tab[2, 2])
SE_logOR
```

```
## [1] 0.0618753
```

```
CI_logOR = logOR + c(-1, 1) * 1.96 * SE_logOR
CI_logOR
```

```
## [1] 0.3791112 0.6216624
```

2. Logistic regress mscd on eversmk. Compare the regression coefficient and its standard error with the log odds ratio and standard error in Part A Question 1 above.

```
fit1 = glm(mscd ~ eversmk, family = "binomial", data = nmes)
summary(fit1)$coefficients
```

```
##               Estimate Std. Error    z value     Pr(>|z|)
## (Intercept) -2.3687101 0.05025572 -47.133141 0.000000e+00
## eversmk      0.5003868 0.06187530   8.087021 6.114183e-16
```

The estimated regression coefficient and confidence interval are the same; they differ only due to differences in rounding.

3. Logistic regress eversmk on mscd. Compare the log odds ratio and standard error from this regression with those from Part A Questions 1 and 2.

```
fit2 = glm(eversmk ~ mscd, family = "binomial", data = nmes)
summary(fit2)$coefficients
```

```
##              Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) 0.2155924 0.01975894 10.911131 1.019801e-27
## mscd        0.5003868 0.06187529  8.087022 6.114134e-16
```

The estimated log odds ratio and confidence interval are the same as when we performed the logistic regression of mscd on eversmk.

4.Write a couple of sentences that can be used to teach a public health professional: the interpretation of the logistic regression coefficient; and the invariance principle of the odds ratio.

In logistic regression analyses, the regression coefficient for a categorical explanatory variable with 2 levels (for instance, ever smoker; 1 = yes, 0 = no) represents the log odds ratio comparing the log odds of the response variable across the two levels of the explanatory variable. For example, consider a logistic regression analysis of having a major smoking-caused disease on being an ever smoker (1 = yes, 0 = no) using the 1987 NMES dataset. The estimated regression coefficient is 0.5, which represents the difference in the log odds of having a major smoking-caused disease comparing ever smokers to never smokers.

Therefore, the estimated odds ratio of having a major smoking-caused disease comparing ever smokers to never smokers is $\exp(0.5) = 1.65$. Ever smokers have estimated odds of having a major smoking-caused disease that are 65% greater than the odds for never smokers. In an analysis with two binary variables (X and Y), the estimated odds ratio for P (X = 1|Y) comparing Y =1 to Y=0 is the same as the estimated odds ratio for P(Y =1|X) comparing X=1 to X=0. Therefore, the estimate of the odds ratio is invariant to the choice of the response variable. That is, either X or Y may be considered the response variable and the other the predictor variable. The log odds ratio estimate will be the same.

5. Extra enjoyment. Review the paper by Prentice and Pyke (Biometrika, 1979) and then state the invariance property of the log odds ratio estimate from a logistic regression in precise mathematical terms.

Again assuming that we have an analysis of two binary variables (X and Y), then the invariance property of the odds ratio can be shown as follows: First note that:

$$P(X=1|Y=1) = \frac{P(Y=1|X=1)P(X=1)}{P(Y=1)}$$

Second, apply the above result to the definition of the odds ratio:

$$OR = \frac{\frac{P(X=1|Y=1)}{P(X=0|Y=1)}}{\frac{P(X=1|Y=0)}{P(X=0|Y=0)}} = \frac{P(Y=1|X=1)/P(Y=1|X=0)}{P(Y=0|X=1)/P(Y=0|X=0)}$$

There are typically two ways in which studies are preformed to associate a binary outcome (such as disease status) with another binary outcome (such as exposure to a risk factor for the disease):

- Prospective study: where you sample persons based on whether or not they have a risk factor (X) and follow them until you observe the disease status (Y)
- Retrospective study: where you sample persons based on whether or not they have the disease (Y) and then observe whether or not they were exposed to the risk factor (X)

We have just shown that in either case, the odds ratio relating the disease (Y) to the risk factor (X) is estimable (via logistic regression!).

Part B. Association of eversmk and mscd, controlling for age.

1. Stratify age by: <50, 51-60, 61-70, >70. Within each stratum, calculate the log odds ratio and standard error for the mscd-eversmk association.

Complete the table below. Here, weight is defined to be the inverse of the variance normalized to sum to 1.0 across strata: $weight_j = (1/se_j^2)/\Sigma_j(1/se_j^2)$.

```
nmes <- nmes %>%
  mutate(agestrat = case_when(lastage <= 50 ~ "<=50",
         (lastage > 50 & lastage <= 60) ~ "51-60",
         (lastage > 60 & lastage <= 70) ~ "61-70",
         (lastage > 70) ~ ">70"))

table(nmes$agestrat)
```

```
##
##  <=50   >70 51-60 61-70
##  3384  2857  2478  2965
```

```
logodds_50 <- summary(glm(mscd~eversmk,data=nmes[nmes$agestrat == "<=50",],
                          family="binomial"))
logodds_60 <- summary(glm(mscd~eversmk,data=nmes[nmes$agestrat == "51-60",],
                          family="binomial"))
logodds_70 <- summary(glm(mscd~eversmk,data=nmes[nmes$agestrat == "61-70",],
                          family="binomial"))
logodds_70_plus <- summary(glm(mscd~eversmk,data=nmes[nmes$agestrat == ">70",],
                               family="binomial"))

age_stratum <- c("<=50", "51-60", "61-70", ">70")
age_logodds <- c(logodds_50$coefficients[2,1], logodds_60$coefficients[2,1],
                 logodds_70$coefficients[2,1], logodds_70_plus$coefficients[2,1])
age_se <- c(logodds_50$coefficients[2,2], logodds_60$coefficients[2,2],
            logodds_70$coefficients[2,2], logodds_70_plus$coefficients[2,2])
```

```
# Calculate inverse variance and weights
tab_byage <- tibble(age_stratum, age_logodds, age_se)
tab_byage$inv <- 1/(tab_byage$age_se^2)
inv_sum <- sum(tab_byage[-5,]$inv)
```

```
## Warning: The `i` argument of ``[.tbl_df`()` must lie in [-rows, 0] if negative, as of tibble 3.0.0.
## Use `NA` as row index to obtain a row full of `NA` values.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
tab_byage$weight <- tab_byage$inv/inv_sum

#
knitr::kable(tab_byage,
      col.names = c("Age Stratum (j)", "Log Odds Ratio",
                    "Std Error (se_j)", "1/(se_j^2)",
                    "Weight (1/se_j^2)/sum_j(1/se_j^2)"))
```

Age Stratum (j)	Log Odds Ratio	Std Error (se_j)	1/(se_j^2)	Weight (1/se_j^2)/sum_j(1/se_j^2)
<=50	0.9245442	0.2686846	13.85206	0.0587162
51-60	0.9194640	0.1948103	26.34973	0.1116915
61-70	0.7291328	0.1141601	76.73110	0.3252485
>70	0.5823418	0.0916766	118.98243	0.5043438

2. Calculate the weighted average log odds ratio from the data above and its standard error.

The standard error = $\sqrt{1/\Sigma_j[1/se_j^2]}$. Compare this value to the one from Part A. Question 1 where age was not controlled by plotting each coefficient with its confidence interval on the same set of axes. Add any additional relevant information for evaluating whether age is a confounder to your figure. Explain in a sentence or two whether age is a "confounder" of the smoking-disease association and why?

```
par(mfrow = c(1, 1))
avg_logOR = sum(tab_byage$age_logodds * tab_byage$weight)
avg_logOR
```

```
## [1] 0.6878318
```

SE of weighted average log OR = sqrt{1/sum[1/Variance]}

```
SE_avg_logOR = sqrt(1/sum(tab_byage$inv))
SE_avg_logOR
```

```
## [1] 0.06510614
```

confidence interval

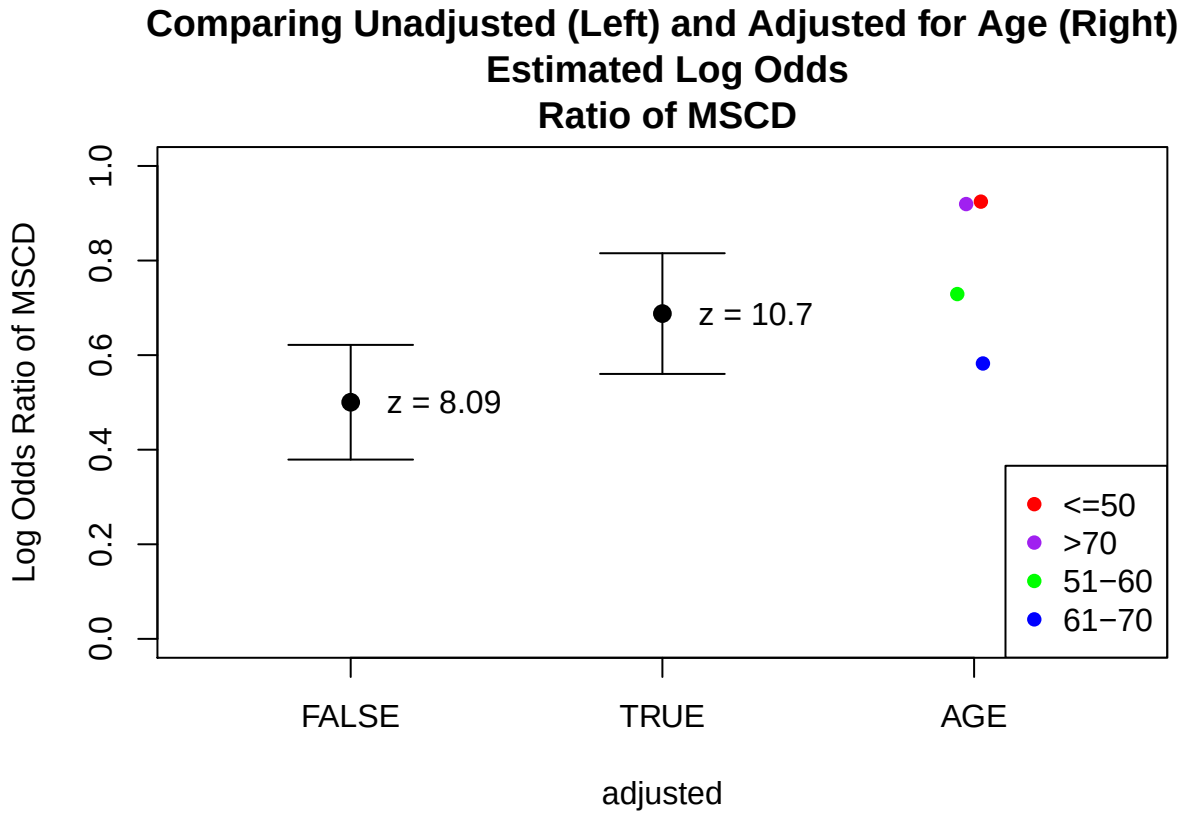
```
CI_avg_logOR = avg_logOR + c(-1, 1) * 1.96 * SE_avg_logOR
CI_avg_logOR
```

```
## [1] 0.5602238 0.8154399
```

Plot adjusted and unadjusted log OR estimates and CIs

```
df_CI = data.frame(adjusted = c(FALSE, TRUE),
        estimate = c(logOR, avg_logOR),
```

```
        LB = c(CI_logOR[1], CI_avg_logOR[1]),
        UB = c(CI_logOR[2], CI_avg_logOR[2]))
age_stratum_col = c("red", "purple", "green", "blue")
fit.temp.1 <- glm(eversmk ~ mscd , family = "binomial", data = nmes)
val.1 <- summary(fit.temp.1)$coefficients[2, 3]
fit.temp.2 <- glm(eversmk ~ mscd + agestrat, family = "binomial", data = nmes)
val.2 <- summary(fit.temp.2)$coefficients[2,3]
plot(c(1, 2),
     c(df_CI$estimate),
     xlab = "adjusted", ylab = "Log Odds Ratio of MSCD",
     main = "Comparing Unadjusted (Left) and Adjusted for Age (Right)\n Estimated Log Odds
 Ratio of MSCD", ylim = c(0, 1), xlim = c(0.5, 3.5), xaxt = "n", pch = 16, cex = 1.3)
segments(x0 = 1, x1 = 1, y0 = df_CI$LB[1], y1 = df_CI$UB[1])
segments(x0 = 2, x1 = 2, y0 = df_CI$LB[2], y1 = df_CI$UB[2])
segments(x0 = 0.8, x1 = 1.2, y0 = df_CI$LB[1], y1 = df_CI$LB[1])
segments(x0 = 0.8, x1 = 1.2, y0 = df_CI$UB[1], y1 = df_CI$UB[1])
segments(x0 = 1.8, x1 = 2.2, y0 = df_CI$LB[2], y1 = df_CI$LB[2])
segments(x0 = 1.8, x1 = 2.2, y0 = df_CI$UB[2], y1 = df_CI$UB[2])
points(jitter(c(3,3,3,3)), tab_byage$age_logodds, col = age_stratum_col, pch = 16)
axis(1, at = c(1, 2, 3), labels = c("FALSE", "TRUE", "AGE"))
legend("bottomright", legend = levels(as.factor(age_stratum)),
       col = age_stratum_col, pch = 16)
text(1.3, df_CI$estimate[1], paste0("z = ", round(val.1, 2)))
text(2.3, df_CI$estimate[2], paste0("z = ", round(val.2, 2)))
```



The age-adjusted log odds ratio is larger than the unadjusted log odds ratio; in addition, each of the

age-stratum specific log odds ratios are larger than the unadjusted log odds ratio AND the test statistic for the age-adjusted log odds ratio is greater than the unadjusted test statistic (10 vs. 8). All of this supports the claim that age is a confounder for the MSCD - eversmoker relationship.

3. Logistic regress mscd on eversmk and 3 indicator variables for the 4 age strata, i.e. make age category a factor.

Compare the resulting eversmk coefficient and standard error with the value above in Part B Question 2.

```
fit3 = glm(mscd ~ eversmk + factor(agestrat), family = "binomial", data = nmes)
summary(fit3)$coefficients
```

```
##                         Estimate Std. Error    z value      Pr(>|z|)
## (Intercept)           -4.1449975 0.12103815 -34.245381 5.109576e-257
## eversmk                0.6906547 0.06451845  10.704763  9.667779e-27
## factor(agestrat)>70    2.5132897 0.12078968  20.807155  3.728226e-96
## factor(agestrat)51-60  1.0473231 0.13725073   7.630729  2.334295e-14
## factor(agestrat)61-70  1.9611610 0.12274999  15.976874  1.852045e-57
```

The eversmk coefficient and standard error are very similar to the one calculated in II.2b.

4. Now repeat the analysis controlling for age with your favorite smooth function of continuous age with three degrees of freedom.

```
fit4 = glm(mscd ~ eversmk + ns(lastage, df = 3), family = "binomial", data = nmes)
# see where knots placed
ns_terms = as.character(attr(terms(fit4), "predvars"))[4]
# extract characters in string that correspond to knot locations (works for 2 knots)
knots = gsub(".+, knots = c.([1-9]+, [1-9]+).+", "\\1", ns_terms)
# use strsplit() to split string into a vector containing both knot # locations
knots = as.numeric(strsplit(knots, split = ", ")[[1]])
knots
```

```
## [1] 53 67
```

```
coef_fit4 = round(coefficients(summary(fit4))[2, ], 3)
coef_fit4
```

```
##   Estimate Std. Error    z value   Pr(>|z|)
##      0.732      0.066     11.153      0.000
```

```
#summary(fit4)
```

5. Regress eversmk on mscd and the same function of age. Compare the mscd coefficient and standard error from this model with the eversmk coefficient from the model in Part B Question 4.

```
fit5 = glm(eversmk ~ mscd + ns(lastage, df = 3), family = "binomial", data = nmes)
# see where knots placed (should be the same as for fit4)
attr(terms(fit5), "predvars")
```

```
## list(eversmk, mscd, ns(lastage, knots = c(`33.33333%` = 53, `66.66667%` = 67
## ), Boundary.knots = c(40L, 94L), intercept = FALSE))
```

```
coef_fit5 = round(coefficients(summary(fit5))[2, ], 3)
coef_fit5
```

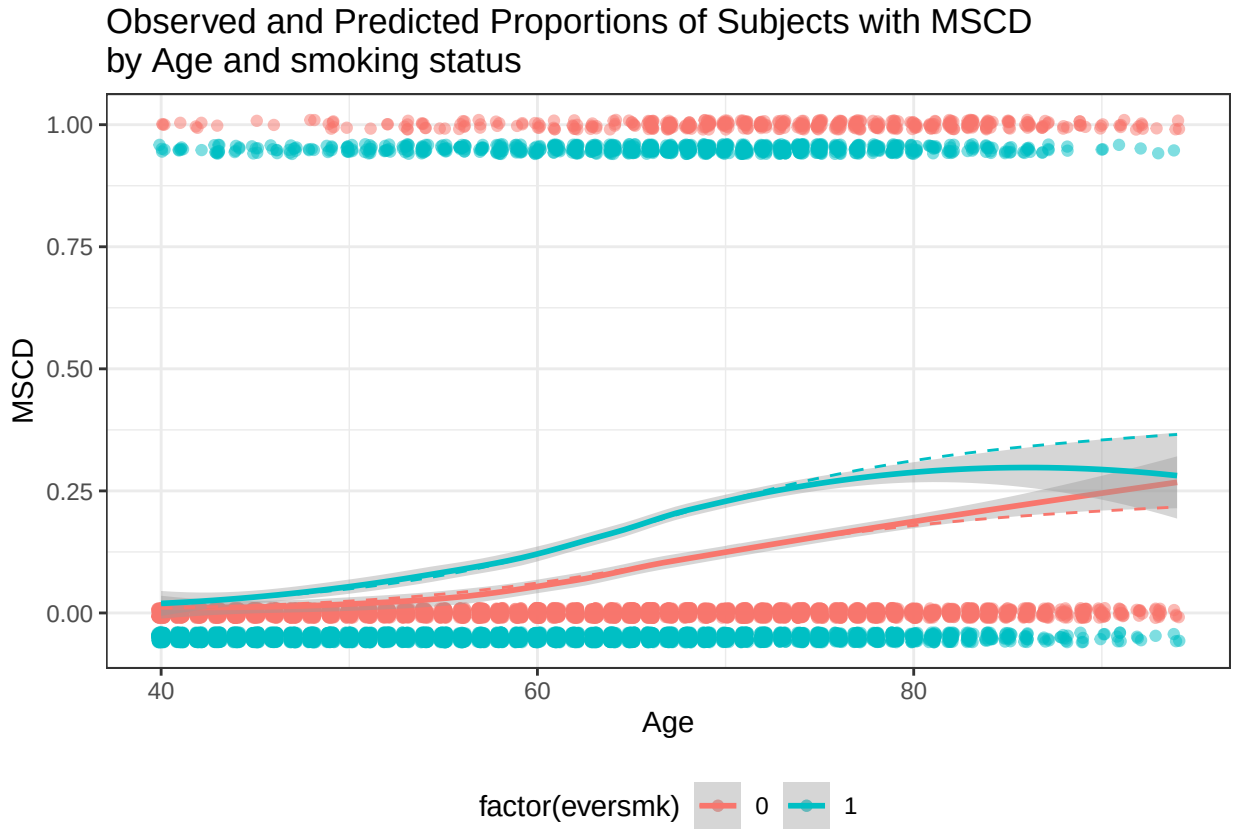
```
##   Estimate Std. Error    z value   Pr(>|z|)
##      0.733      0.066     11.170      0.000
```

```
#summary(fit5)
```

For the regression of MSCD on eversmk, the estimate (standard error) for the coefficient for eversmk is 0.732 (0.066); for the regression of eversmk on MSCD, the estimate (standard error) for the coefficient for MSCD is 0.733 (0.066). The two coefficients and standard errors are very similar.

6. Plot the mscd data against age using the eversmk value as the plotting symbol or color. Add the predicted values from your model and a kernel smoother fit separately to each smoking group for comparison. Compare the model predictions with the kernel smoothers to see if there is evidence of effect modification of the smoking-mscd association by age?

```
# generate predictions on probability scale
nmes$pred = expit(predict(fit4))
# plot observed probabilities using loess smoother, by eversmk plot
# predicted probabilities, by eversmk jitter observations and separate
# observations vertically by eversmk
ggplot(nmes, aes(x = lastage, y = mscd * 1, colour = factor(eversmk))) +
  geom_jitter(data = filter(nmes, eversmk == 0),
  position = position_jitter(width = 0.2 , height = 0.01), alpha = 0.5) +
  geom_jitter(data = filter(nmes, eversmk == 1), aes(y = mscd * 1 - 0.05),
  position = position_jitter(width = 0.2, height = 0.01), alpha = 0.5) +
  geom_smooth(aes(group = eversmk), method = "loess") +
  geom_line(aes(y = pred, group = eversmk), linetype = 2) +
  theme_bw() +
  theme(legend.position = "bottom") +
  xlab("Age") +
  ylab("MSCD") +
  ggtitle("Observed and Predicted Proportions of Subjects with MSCD \nby Age and smoking status")
```



The solid lines and confidence bands show the smoothed observed proportions of subjects with MSCD by age for smokers and non-smokers. The dotted lines show the fitted proportions based on a logistic regression model of MSCD against eversmk and a smooth function of age. The smoothed proportions are very similar to the predicted probabilities for ages below 75. The two curves deviate for the older ages. Note that the two observed curves begin to taper together as age approaches the upper boundary. Since we fit a regression model with a fixed effect for eversmk but no interaction effect, the model is unable to predict this modification of the age-MSCD relationship by ever smoking status. In addition, there is relatively little data for persons over 75, reflected in the wide confidence bands for the observed curves. Therefore, we conclude that while the predicted curves reflect the observed curves reasonably well, we cannot reject the possibility that eversmk modifies the age- MSCD relationship.

7. Propose an extended model to directly address the possibility that age modifies the effect of smoking on disease prevalence. Fit this model and compare it to the model without effect modification using a likelihood ratio test.

```
fit6 = glm(mscd ~ eversmk * ns(lastage, df = 3), family = "binomial", data = nmes)
#summary(fit6)$coeff
```

```
lrtest(fit4,fit6)
```

```
## Likelihood ratio test
##
## Model 1: mscd ~ eversmk + ns(lastage, df = 3)
## Model 2: mscd ~ eversmk * ns(lastage, df = 3)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   5 -3669.7
## 2   8 -3666.7  3 6.0825     0.1077
```

The p-value from the likelihood ratio test is 0.11, so there is not enough statistical evidence to reject the null hypothesis of no effect modification.

Part III: Now you are the course instructor!

Using the data and analyses you did in this homework as an example, prepare an .Rmd file and accompanying vignette (written or audio and/or video) that can be used to teach logistic regression to a public health professional. You can work in groups on the vignette. Be numerate, avoid non-essential statistical jargon and be as clear as possible. Remember to emphasize: Question, Question, Question.

Solutions will vary here; we will post a few selected solutions after grading is complete.