1 Math

Let X be the binary treatment indicator, and M be the continuous effect modifier. The logistic regression model is defined as follows.

$$
\begin{aligned}
g(E[Y]) &= \beta_0 + \beta_1 X + \beta_2 M + \beta_3 MX + \beta_4 C \\
&\text{where } g(\cdot) \text{ is the logit function} \\
\\
&\text{By transformation} \\
g(E[Y]) &= \beta_0 + (\beta_1 X + \beta_3 MX) + \beta_2 M + \beta_4 C \\
&= \beta_0 + (\beta_1 + \beta_3 M)X + \beta_2 M + \beta_4 C
\end{aligned}
$$

Thus, the log OR of interest is $(\beta_1 + \beta_3 M)$, a function of M, *i.e.,* it varies depending on the value of M. The estimate of this log OR, $(\hat{\beta}_1 + \hat{\beta}_3 M)$, and its standard error, $\sqrt{Var(\hat{\beta}_1 + \hat{\beta}_3 M)}$ are also functions of M. The functional form of $Var(\hat{\beta}_1 + \hat{\beta}_3 M)$ is the following.

$$
\begin{aligned}
Var(\hat{\beta}_1 + \hat{\beta}_3 M) &= Var(\hat{\beta}_1) + Var(\hat{\beta}_3 M) + 2Cov(\hat{\beta}_1, \hat{\beta}_3 M) \\
&= Var(\hat{\beta}_1) + (M^2)Var(\hat{\beta}_3) + 2(M)Cov(\hat{\beta}_1, \hat{\beta}_3) \\
&= Var(\hat{\beta}_3)(M^2) + 2Cov(\hat{\beta}_1, \hat{\beta}_3)(M) + Var(\hat{\beta}_1) \\
f(M) &= Var(\hat{\beta}_3)(M^2) + 2Cov(\hat{\beta}_1, \hat{\beta}_3)(M) + Var(\hat{\beta}_1) \\
&\text{Estimated by} \\
\hat{f}(M) &= \widehat{Var}(\hat{\beta}_3)(M^2) + 2\widehat{Cov}(\hat{\beta}_1, \hat{\beta}_3)(M) + \widehat{Var}(\hat{\beta}_1)
\end{aligned}
$$

This can be obtained from R's `vcov(model)` function.

2 Implementation in R

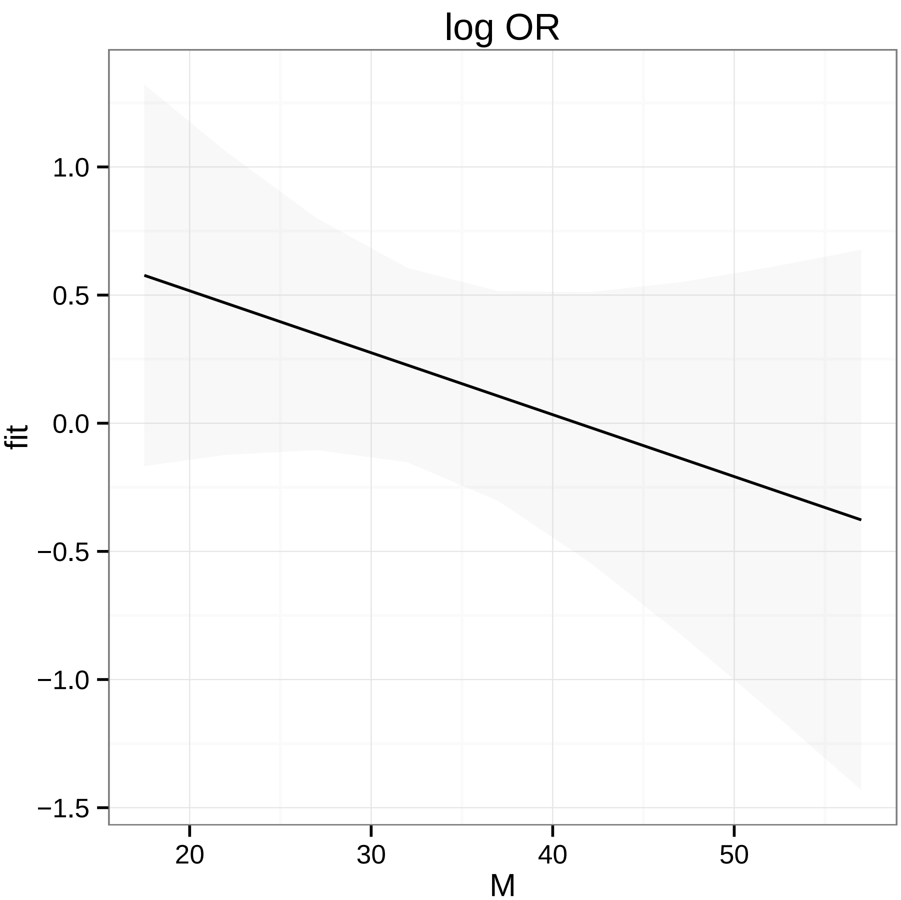
```
library(AER)
data(Affairs)
summary(Affairs)

##     affairs          gender         age         yearsmarried    children
##  Min.   : 0.000   female:315   Min.   :17.50   Min.   : 0.125   no :171
##  1st Qu.: 0.000   male  :286   1st Qu.:27.00   1st Qu.: 4.000   yes:430
##  Median : 0.000                Median :32.00   Median : 7.000
##  Mean   : 1.456                Mean   :32.49   Mean   : 8.178
##  3rd Qu.: 0.000                3rd Qu.:37.00   3rd Qu.:15.000
##  Max.   :12.000                Max.   :57.00   Max.   :15.000
##  religiousness     education       occupation         rating
##  Min.   :1.000   Min.   : 9.00   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:14.00   1st Qu.:3.000   1st Qu.:3.000
##  Median :3.000   Median :16.00   Median :5.000   Median :4.000
##  Mean   :3.116   Mean   :16.17   Mean   :4.195   Mean   :3.932
##  3rd Qu.:4.000   3rd Qu.:18.00   3rd Qu.:6.000   3rd Qu.:5.000
##  Max.   :5.000   Max.   :20.00   Max.   :7.000   Max.   :5.000

library(dplyr)
## Rename variables to be consistent with the math above
Affairs <- within(Affairs,{
                  Y <- as.numeric(affairs > 0)
                  X <- as.numeric(gender == "male")
                  M <- age
                  C <- as.numeric(children == "yes")
              })
```

```r
## Logistic regression
resLogit <- glm(formula = Y ~ X + M + M:X + C,
                family  = binomial(link = "logit"),
                data    = Affairs)
```

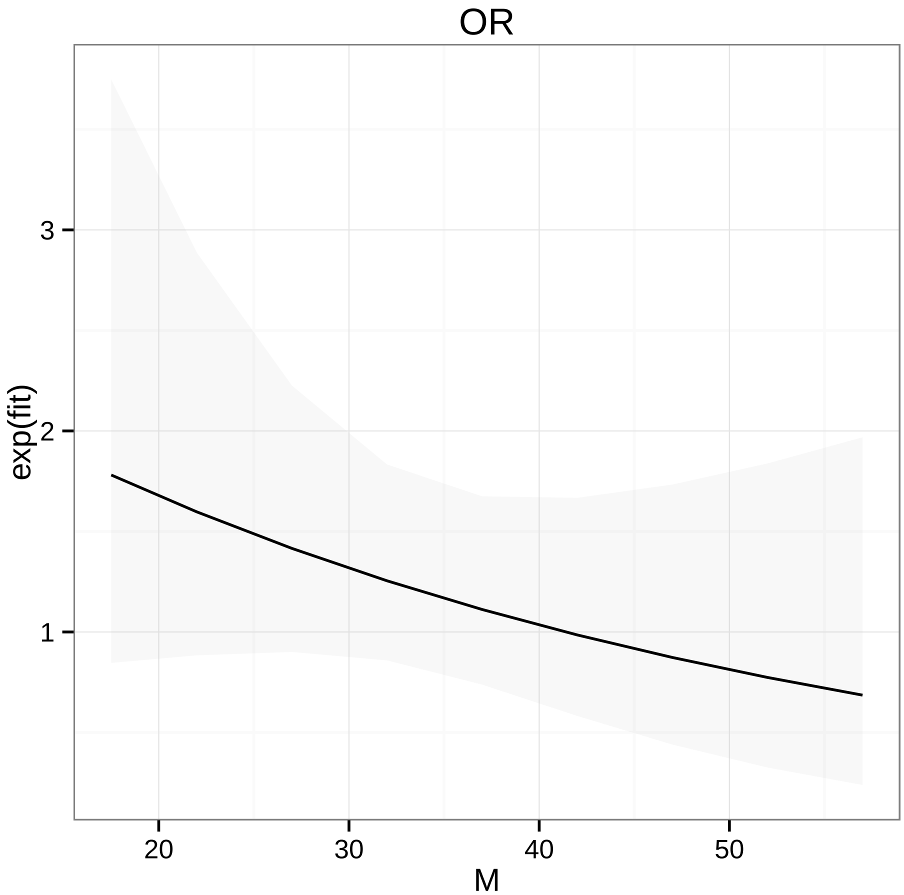
2.1 DIY method for log OR and OR

```r
## Check variance covariance matrix
vcov(resLogit)

##               (Intercept)            X             M             C
## (Intercept)  0.262133214 -0.259085808 -0.0073865340 -0.0089366873
## X           -0.259085808  0.520335035  0.0078352248 -0.0133010409
## M           -0.007386534  0.007835225  0.0002626842 -0.0013158109
## C           -0.008936687 -0.013301041 -0.0013158109  0.0652133820
## X:M          0.007507364 -0.014610444 -0.0002448935  0.0004340834
##                       X:M
## (Intercept)  0.0075073638
## X           -0.0146104440
## M           -0.0002448935
## C            0.0004340834
## X:M          0.0004415789

## Get variance estimates
hatVarBeta1      <- vcov(resLogit)["X","X"]
hatVarBeta3      <- vcov(resLogit)["X:M","X:M"]
hatCovBeta1Beta3 <- vcov(resLogit)["X","X:M"]
## Get coefficient estimates
hatBeta1 <- coef(resLogit)["X"]
hatBeta3 <- coef(resLogit)["X:M"]
## Function to estimate the variance of OR depending on M
varByM <- function(M) {
    hatVarBeta3*M^2 + 2*hatCovBeta1Beta3*M + hatVarBeta1
}
## Function to estimate the log OR depending on M
logOrByM <- function(M) {
    hatBeta3*M + hatBeta1
}
## Create a dataset to predict for
newdat <- expand.grid(X = 1,
                      M = quantile(Affairs$M, probs = seq(0,1,0.01)),
                      C = 0)
newdat$fit     <- logOrByM(newdat$M)
newdat$se.fit  <- varByM(newdat$M) %>% sqrt
## 95% CI
newdat$lower <- newdat$fit - 1.96 * newdat$se.fit
newdat$upper <- newdat$fit + 1.96 * newdat$se.fit
## Check data
head(newdat, 10)

##    X    M C       fit    se.fit      lower    upper
## 1  1 17.5 0 0.5771131 0.3797407 -0.1671787 1.321405
## 2  1 22.0 0 0.4683863 0.3019928 -0.1235197 1.060292
## 3  1 22.0 0 0.4683863 0.3019928 -0.1235197 1.060292
## 4  1 22.0 0 0.4683863 0.3019928 -0.1235197 1.060292
## 5  1 22.0 0 0.4683863 0.3019928 -0.1235197 1.060292
## 6  1 22.0 0 0.4683863 0.3019928 -0.1235197 1.060292
## 7  1 22.0 0 0.4683863 0.3019928 -0.1235197 1.060292
## 8  1 22.0 0 0.4683863 0.3019928 -0.1235197 1.060292
## 9  1 22.0 0 0.4683863 0.3019928 -0.1235197 1.060292
## 10 1 22.0 0 0.4683863 0.3019928 -0.1235197 1.060292
```

```r
## Plot log OR
library(ggplot2)
plotSkeleton <- ggplot(data = newdat, mapping = aes(x = M, y = fit, ymin = lower, ymax = upper)) +
    layer(geom = "ribbon", fill = "gray", alpha = 0.1) +
    layer(geom = "line") +
    theme_bw() + theme(legend.key = element_blank())
plotSkeleton + labs(title = "log OR")
```



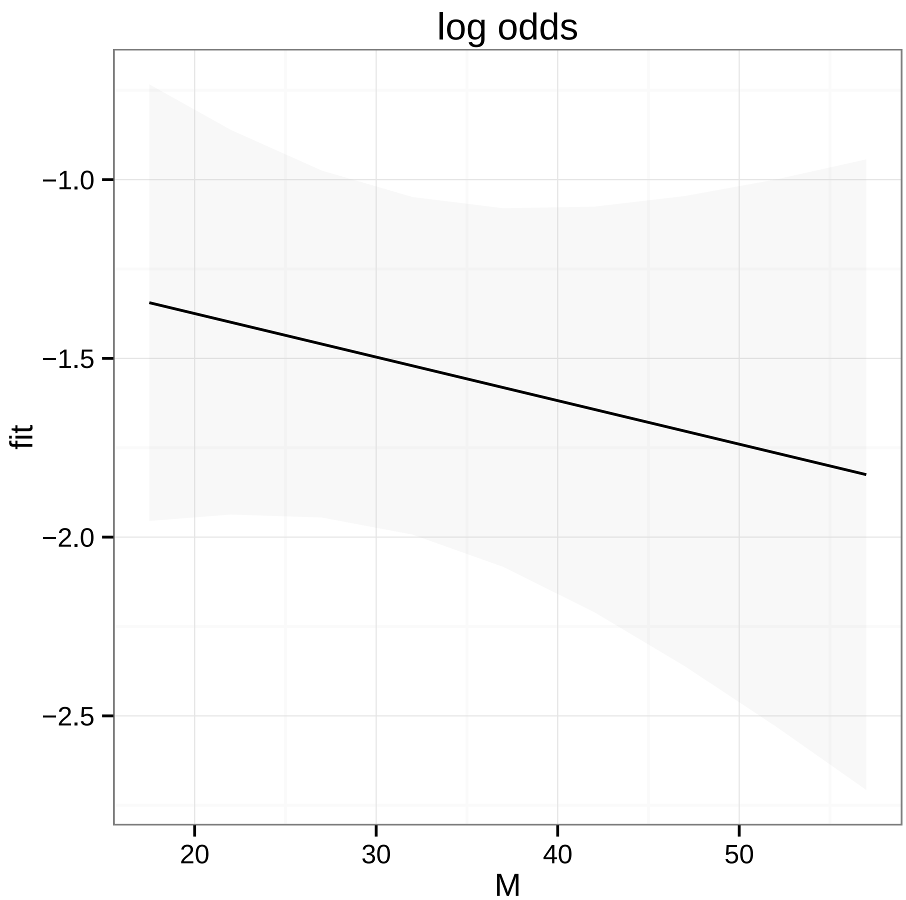

```r
## Plot OR
plotSkeleton + aes(y = exp(fit), ymin = exp(lower), ymax = exp(upper)) + labs(title = "OR")
```



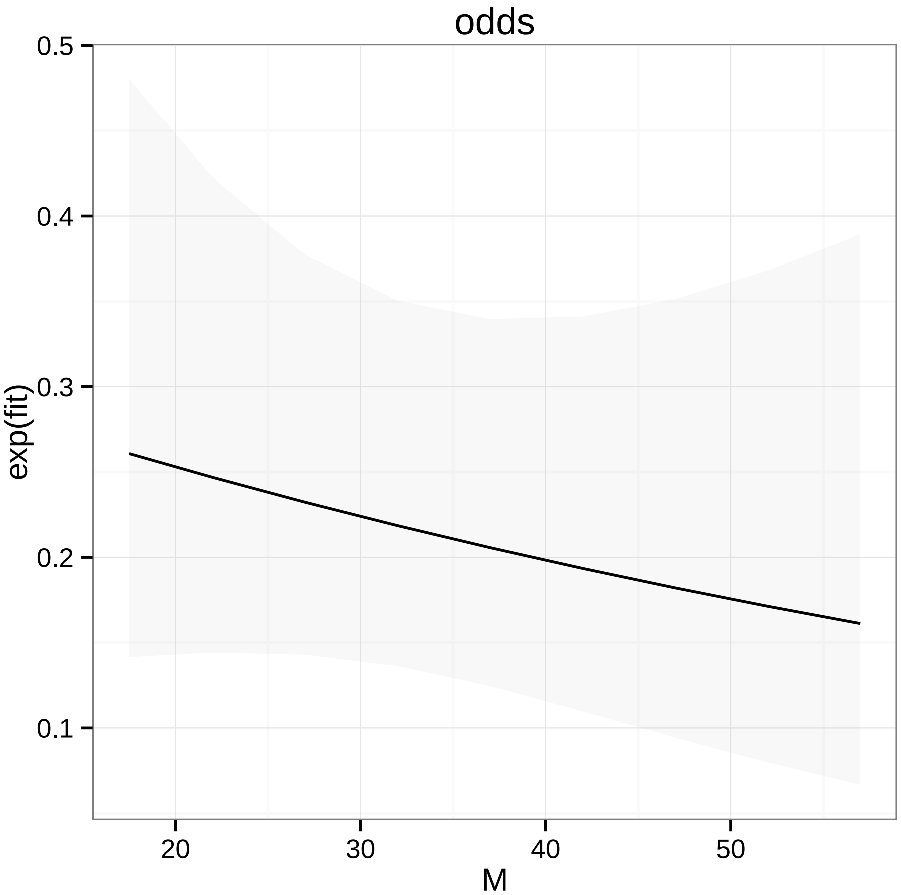
2.2 Using predict to obtain log odds and odds

These are log odds and odds, *i.e.* absolute measures, not effect estimates of X.

```
## By using predict
newdatPredict <- predict(resLogit, newdata = newdat, se.fit = TRUE)
newdatPredict <- newdatPredict[c("fit","se.fit")] %>% data.frame
newdatPredict$M <- newdat$M
## 95% CI
newdatPredict$lower <- newdatPredict$fit - 1.96 * newdatPredict$se.fit
newdatPredict$upper <- newdatPredict$fit + 1.96 * newdatPredict$se.fit
## Plot log odds
plotSkeleton %+% newdatPredict + labs(title = "log odds")
```




```
## Plot odds
plotSkeleton %+% newdatPredict + aes(y = exp(fit), ymin = exp(lower), ymax = exp(upper)) + labs(title = "odds")
```


odds
exp(fit)
0.5
0.4
0.3
0.2
0.1
20
30
40
50
M