# Logistic Regression I

Mike Strube

November 13, 2018

## 1 Preliminaries

***In this section, the RStudio workspace and console panes are cleared of old output, variables, and other miscellaneous debris. Packages are loaded and any required data files are retrieved.***

```
options(replace.assign = TRUE, width = 65, digits = 4, scipen = 4, fig.width = 4,
    fig.height = 4)
# Clear the workspace and console.
rm(list = ls(all = TRUE))
cat("\f")
```

```
# Turn off showing of significance asterisks.
options(show.signif.stars = F)
# Set the contrast option; important for ANOVAs.
options(contrasts = c("contr.sum", "contr.poly"))
how_long <- Sys.time()
set.seed(123)
library(knitr)
```

```
library(psych)
library(ggplot2)

## Warning:  package 'ggplot2' was built under R version 3.4.4
##
## Attaching package:  'ggplot2'
## The following objects are masked from 'package:psych':
##
##     %+%, alpha

library(MASS)
library(sciplot)
library(plyr)
library(aod)
library(MVN)

## Warning:  package 'MVN' was built under R version 3.4.4
## sROC 0.1-2 loaded

library(boot)
```

```
##
## Attaching package:  'boot'
## The following object is masked from 'package:psych':
##
##     logit

library(car)

##
## Attaching package:  'car'
## The following object is masked from 'package:boot':
##
##     logit
## The following object is masked from 'package:psych':
##
##     logit

library(LogisticDx)
library(GGally)

## Warning:  package 'GGally' was built under R version 3.4.4

library(reshape2)
library(MVN)
library(qqplotr)

## Warning:  package 'qqplotr' was built under R version 3.4.4
##
## Attaching package:  'qqplotr'
## The following objects are masked from 'package:ggplot2':
##
##     stat_qq_line, StatQqLine

library(gridExtra)
library(caret)

## Warning:  package 'caret' was built under R version 3.4.4
## Loading required package:  lattice
##
## Attaching package:  'lattice'
## The following object is masked from 'package:boot':
##
##     melanoma
```

# 2 Data

***In this hypothetical example, data from 500 graduate students seeking jobs were examined. Available for each student were three predictors: GRE(V+Q), Years to Finish the Degree, and Number of Publications. The outcome measure was categorical: "Got a job" versus "Did not get a job."***

```
setwd("C:\\Courses\\Psychology 516\\PowerPoint\\2018")

Job <- read.table("jobs_example_for_ppt.csv", sep = ",", header = TRUE)
Job <- as.data.frame(Job)
```

```r
Job$job_result[Job$job == "0"] <- "No Job"
Job$job_result[Job$job == "1"] <- "Job"

# Dummy code for sex.
Job$sex_D <- ifelse(Job$sex == 2, 1, 0)

# Dummy codes for men and women
Job$M_D <- ifelse(Job$sex == 1, 1, 0)
Job$F_D <- ifelse(Job$sex == 2, 1, 0)

# Centered predictors.
Job$gre_c <- as.numeric(scale(Job$gre, scale = FALSE))
Job$pubs_c <- as.numeric(scale(Job$pubs, scale = FALSE))
Job$years_c <- as.numeric(scale(Job$years, scale = FALSE))

# Residuals
Job$gre_R <- lm(gre ~ as.factor(job), data = Job)$residuals
Job$pubs_R <- lm(pubs ~ as.factor(job), data = Job)$residuals
Job$years_R <- lm(years ~ as.factor(job), data = Job)$residuals

describe(Job[, c(3:5, 7, 11:15)])
```

```
##         vars   n    mean     sd  median trimmed    mad     min
## gre        1 500 1296.82 103.72 1297.00 1296.34 102.30 1034.00
## pubs       2 500    4.30   2.31    4.00    4.31   2.97    0.00
## years      3 500    6.09   2.05    6.00    5.82   2.97    4.00
## job        4 500    0.27   0.45    0.00    0.22   0.00    0.00
## M_D        5 500    0.38   0.48    0.00    0.34   0.00    0.00
## F_D        6 500    0.62   0.48    1.00    0.66   0.00    0.00
## gre_c      7 500    0.00 103.72    0.18   -0.48 102.30 -262.82
## pubs_c     8 500    0.00   2.31   -0.30    0.01   2.97   -4.30
## years_c    9 500    0.00   2.05   -0.09   -0.27   2.97   -2.09
##             max range  skew kurtosis   se
## gre     1560.00   526  0.06    -0.34 4.64
## pubs      10.00    10 -0.01    -0.65 0.10
## years     14.00    10  0.90     0.15 0.09
## job        1.00     1  1.02    -0.96 0.02
## M_D        1.00     1  0.51    -1.74 0.02
## F_D        1.00     1 -0.51    -1.74 0.02
## gre_c    263.18   526  0.06    -0.34 4.64
## pubs_c     5.70    10 -0.01    -0.65 0.10
## years_c    7.91    10  0.90     0.15 0.09
```

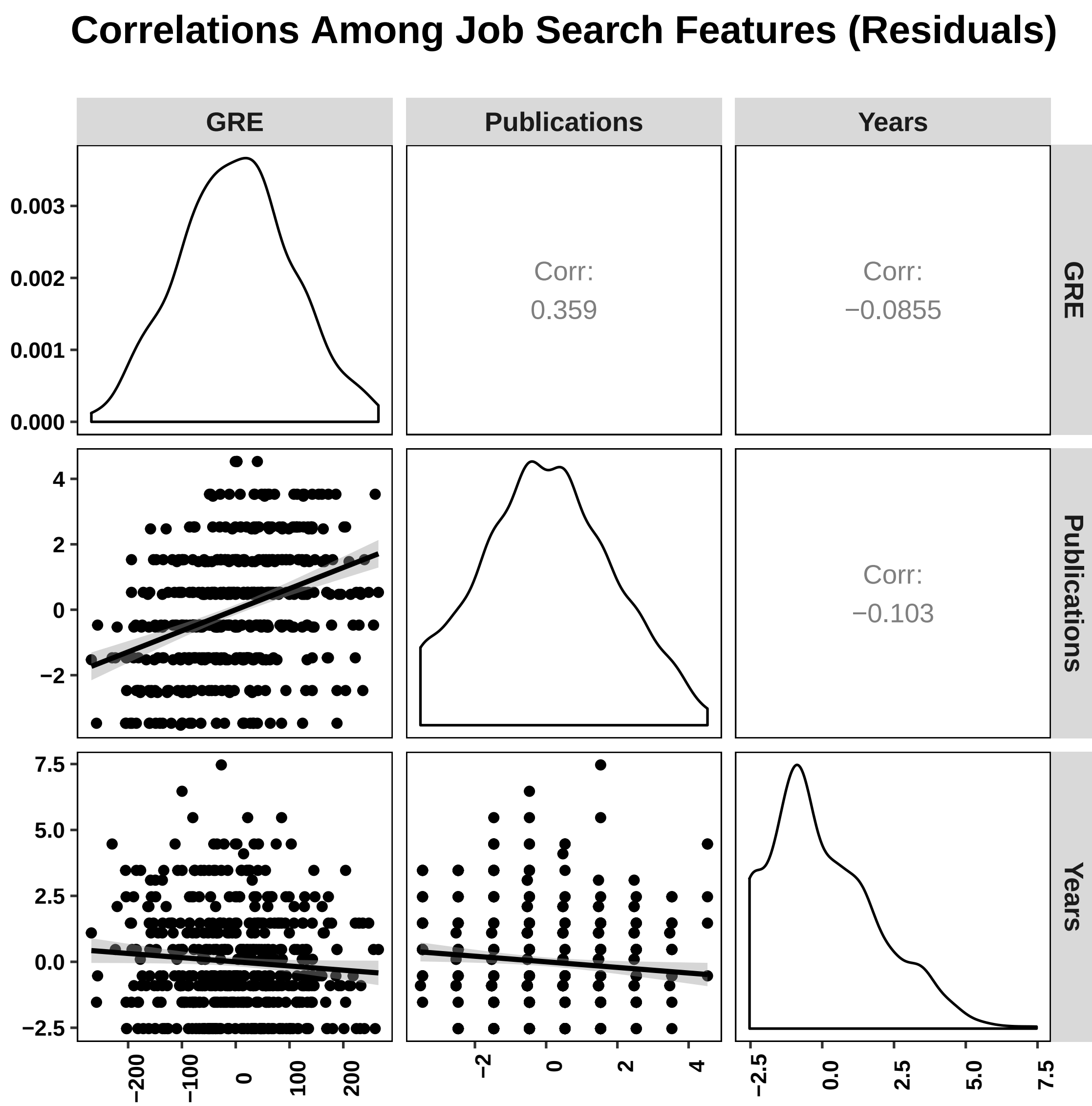
# 3 Job Search Data

***These hypothetical data simulate the factors that might contribute to successfully getting an academic job.***

# 4 Basic Visualization

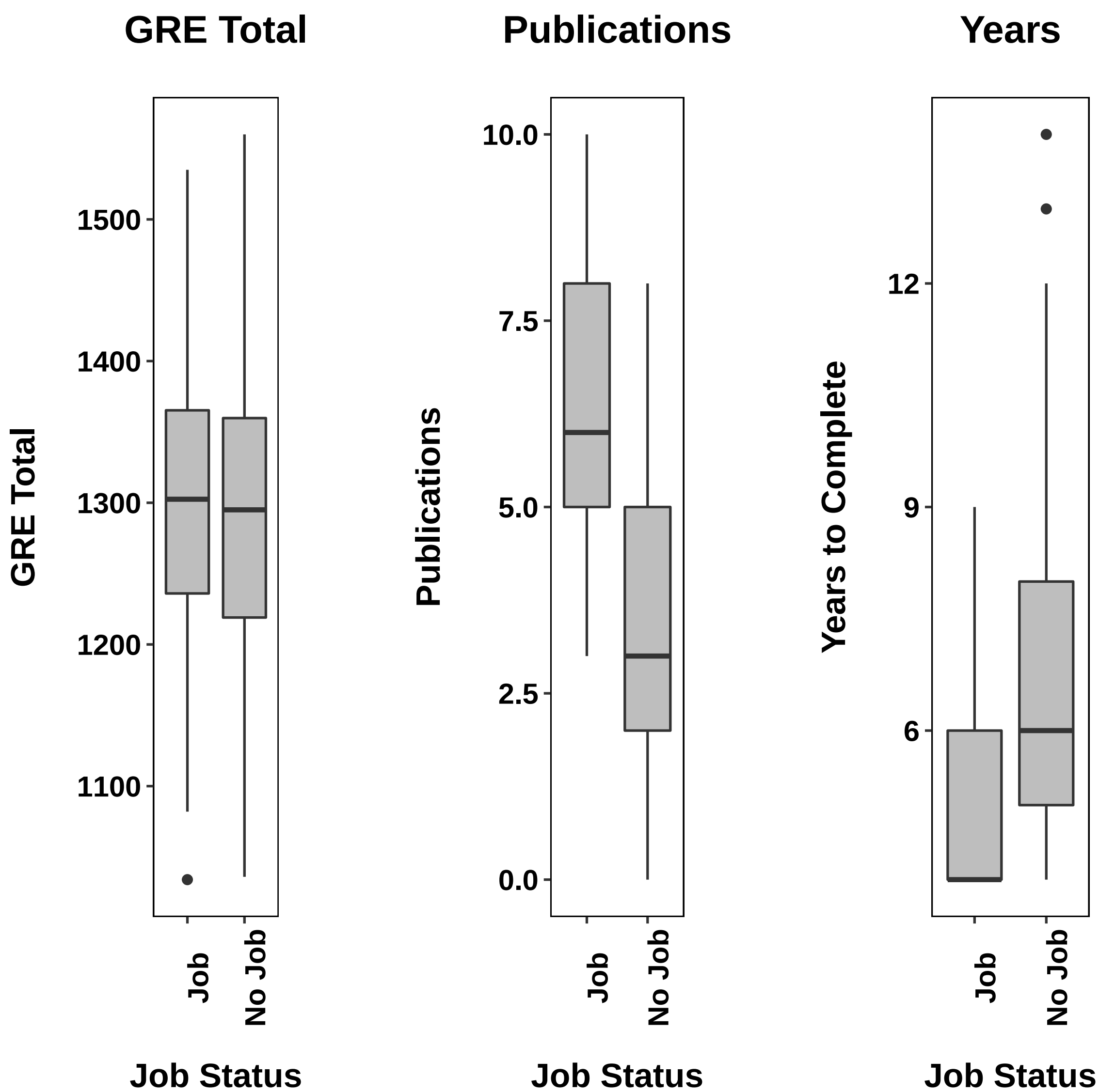
***The basic nature of the data is easily viewed with some simple graphics.***

```
ggpairs(Job[16:18], lower = list(continuous = "smooth"), upper = list(continuous = "cor"),
    columnLabels = c("GRE", "Publications", "Years")) + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 9, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 9, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Correlations Among Job Search Features (Residuals)")
```



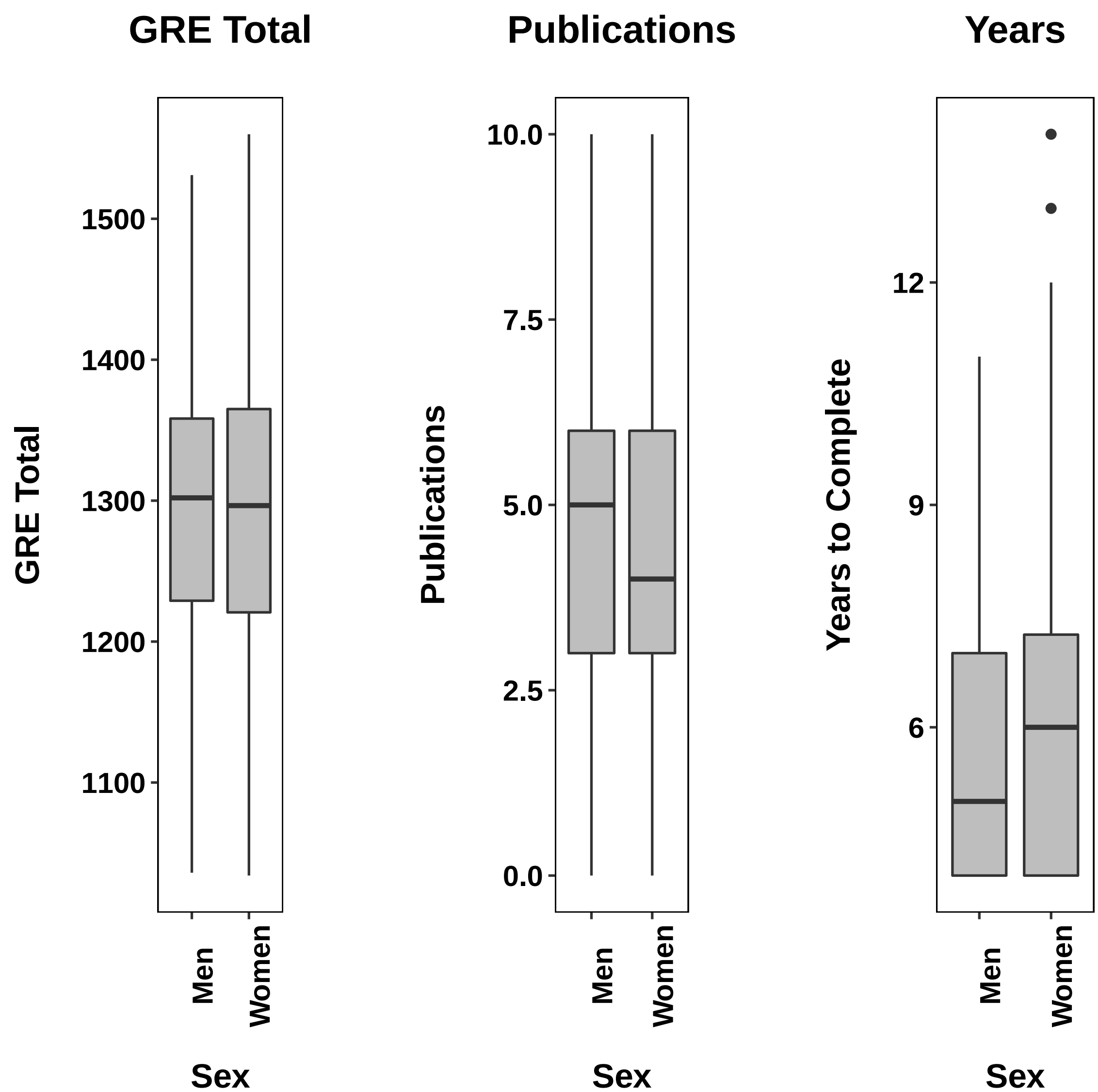

```
p1 <- ggplot(Job, aes(x = job_result, y = gre)) + geom_boxplot(fill = "gray") +
    ylab("GRE Total") + xlab("Job Status") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("GRE Total")
```

```r
p2 <- ggplot(Job, aes(x = job_result, y = pubs)) + geom_boxplot(fill = "gray") +
    ylab("Publications") + xlab("Job Status") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Publications")

p3 <- ggplot(Job, aes(x = job_result, y = years)) + geom_boxplot(fill = "gray") +
    ylab("Years to Complete") + xlab("Job Status") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Years")
grid.arrange(p1, p2, p3, nrow = 1)
```




```
Job$sex_F <- factor(Job$sex, levels = c(1, 2), labels = c("Men", "Women"))
p1 <- ggplot(Job, aes(x = sex_F, y = gre)) + geom_boxplot(fill = "gray") +
    ylab("GRE Total") + xlab("Sex") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("GRE Total")
```

```r
p2 <- ggplot(Job, aes(x = sex_F, y = pubs)) + geom_boxplot(fill = "gray") +
    ylab("Publications") + xlab("Sex") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Publications")

p3 <- ggplot(Job, aes(x = sex_F, y = years)) + geom_boxplot(fill = "gray") +
    ylab("Years to Complete") + xlab("Sex") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Years")
grid.arrange(p1, p2, p3, nrow = 1)
```




# 5 Group Differences

*A univariate look at the data will provide some clues about likely variables of influence in the logistic regression.*

```
Job_MANOVA_1 <- manova(as.matrix(Job[, 3:5]) ~ Job$job)
summary(Job_MANOVA_1)

##              Df Pillai approx F num Df den Df Pr(>F)
## Job$job       1   0.41      115      3    496 <2e-16
## Residuals 498

summary.aov(Job_MANOVA_1)
```

```
##  Response gre :
##               Df  Sum Sq Mean Sq F value Pr(>F)
## Job$job       1    5693    5693    0.53   0.47
## Residuals   498 5362834   10769
##
##  Response pubs :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Job$job       1    927     927     266 <2e-16
## Residuals   498   1733       3
##
##  Response years :
##               Df Sum Sq Mean Sq F value Pr(>F)
## Job$job       1    263   262.6    70.9  4e-16
## Residuals   498   1844     3.7
```

```
table_1 <- table(Job[c("job_result", "sex_F")])
colnames(table_1) <- c("Men", "Women")
row.names(table_1) <- c("Job", "No Job")
table_1
```

```
##           sex_F
## job_result Men Women
##     Job     62    74
##     No Job 126   238
```

```
p_table_1 <- prop.table(table(Job[c("job_result", "sex_F")]), 2)
colnames(p_table_1) <- c("Men", "Women")
row.names(table_1) <- c("Job", "No Job")
p_table_1
```

```
##           sex_F
## job_result    Men  Women
##     Job    0.3298 0.2372
##     No Job 0.6702 0.7628
```

```
chisq.test(table_1)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity
##  correction
##
## data:  table_1
## X-squared = 4.6, df = 1, p-value = 0.03
```

```
Job_MANOVA_2 <- manova(as.matrix(Job[, 3:5]) ~ Job$sex_F)
summary(Job_MANOVA_2)
```

```
##             Df Pillai approx F num Df den Df Pr(>F)
## Job$sex_F   1 0.0159     2.67      3    496  0.047
## Residuals 498
```

```
summary.aov(Job_MANOVA_2)
```

```
##  Response gre :
##               Df  Sum Sq Mean Sq F value Pr(>F)
## Job$sex_F     1    1810    1810    0.17   0.68
```

```
## Residuals   498 5366717   10777
##
##  Response pubs :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Job$sex_F     1     11   11.19     2.1   0.15
## Residuals   498   2648    5.32
##
##  Response years :
##              Df Sum Sq Mean Sq F value Pr(>F)
## Job$sex_F     1     28   27.62    6.61   0.01
## Residuals   498   2079    4.18
```

# 6 Basic Binary Logistic Regression

*A binary logistic regression model is the alternative to discriminant analysis for these data. It is potentially more flexible and also does not have assumptions that are as restrictive. This is just one example of a generalized linear model, so we need to indicate the distribution family and link function that we want.*

*We use a dummy code for sex, with men = 0 and women = 1. This will produce an intercept that is the expected logit for men. For the other predictors, we use centered versions. The intercept will then be the grand mean expected logit in models that only contain continuous predictors. The centered predictors have the usual advantages of reducing multicollinearity in models with product variables.*

## 6.1 Single Predictor Models

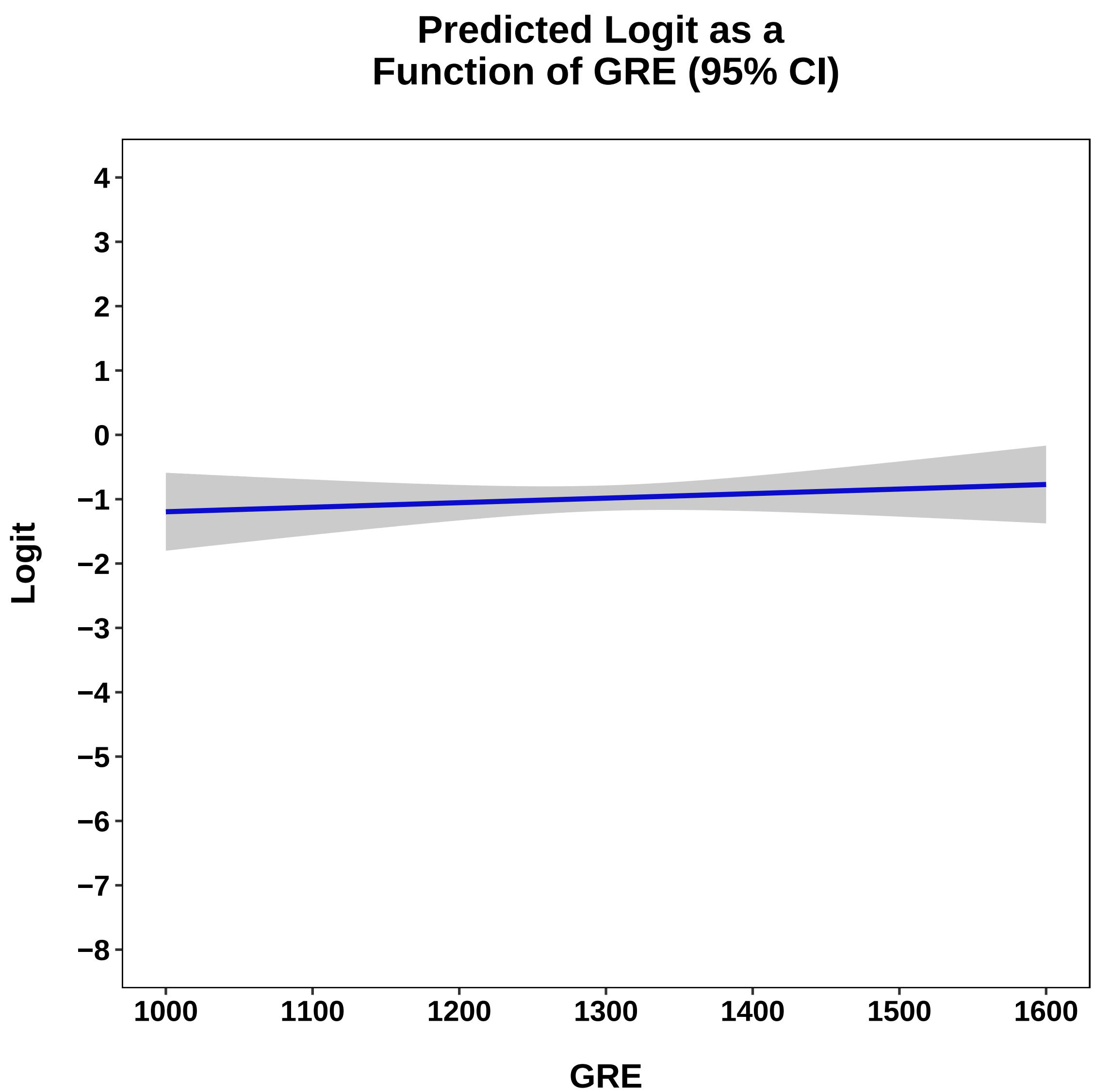
```
Job_BLR_1 <- glm(job ~ sex_D, family = binomial("logit"), data = Job)
summary(Job_BLR_1)

##
## Call:
## glm(formula = job ~ sex_D, family = binomial("logit"), data = Job)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -0.895  -0.895  -0.736   1.490   1.696
##
## Coefficients:
##             Estimate Std. Error z value  Pr(>|z|)
## (Intercept)   -0.709      0.155   -4.57 0.0000048
## sex_D         -0.459      0.204   -2.25     0.025
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 585.24  on 499  degrees of freedom
## Residual deviance: 580.23  on 498  degrees of freedom
## AIC: 584.2
##
## Number of Fisher Scoring iterations: 4

confint(Job_BLR_1)

## Waiting for profiling to be done...

##               2.5 %  97.5 %
## (Intercept) -1.0191 -0.4098
## sex_D       -0.8597 -0.0574

confint.default(Job_BLR_1)

##               2.5 %   97.5 %
## (Intercept) -1.0132 -0.40510
## sex_D       -0.8597 -0.05844

exp(cbind(OR = coef(Job_BLR_1), confint(Job_BLR_1)))
```

```
## Waiting for profiling to be done...

##                 OR  2.5 % 97.5 %
## (Intercept) 0.4921 0.3609 0.6638
## sex_D       0.6319 0.4233 0.9442
```

```
Job_BLR_2 <- glm(job ~ gre_c, family = binomial("logit"), data = Job)
summary(Job_BLR_2)

##
## Call:
## glm(formula = job ~ gre_c, family = binomial("logit"), data = Job)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -0.862  -0.806  -0.784   1.567   1.698
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.985718   0.100609   -9.80   <2e-16
## gre_c        0.000706   0.000970    0.73     0.47
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 585.24  on 499  degrees of freedom
## Residual deviance: 584.71  on 498  degrees of freedom
## AIC: 588.7
##
## Number of Fisher Scoring iterations: 4

confint(Job_BLR_2)

## Waiting for profiling to be done...

##                 2.5 %    97.5 %
## (Intercept) -1.186059 -0.791334
## gre_c       -0.001196  0.002615

confint.default(Job_BLR_2)

##                 2.5 %    97.5 %
## (Intercept) -1.182908 -0.788529
## gre_c       -0.001196  0.002608

exp(cbind(OR = coef(Job_BLR_2), confint(Job_BLR_2)))

## Waiting for profiling to be done...

##                 OR  2.5 % 97.5 %
## (Intercept) 0.3732 0.3054 0.4532
## gre_c       1.0007 0.9988 1.0026
```

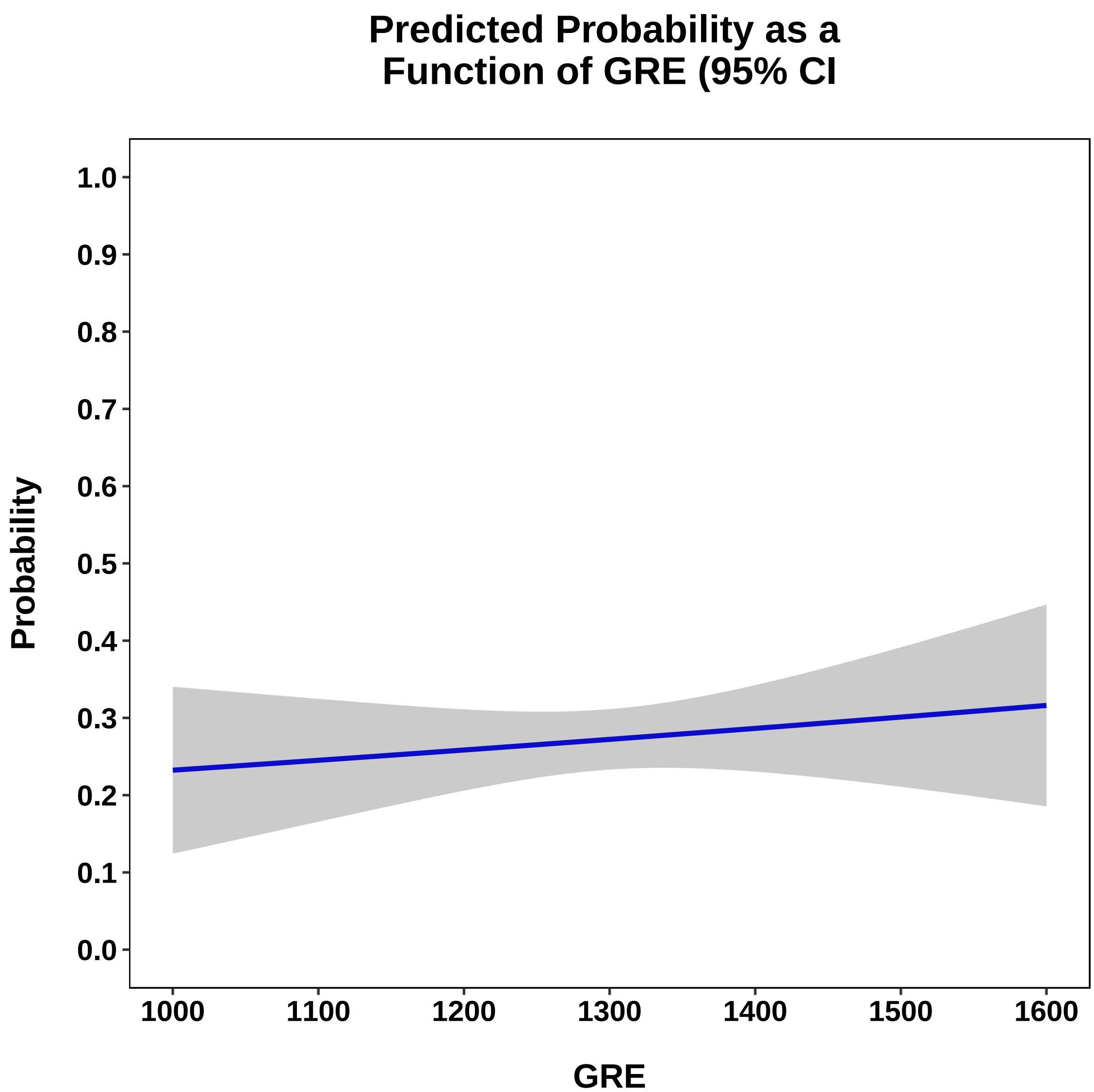
```
predict_data = with(Job, data.frame(gre_c = seq(1000 - mean(Job$gre),
    1600 - mean(Job$gre), 1)))
```

```r
plot_data_p <- predict(Job_BLR_2, predict_data, type = "response",
    se.fit = TRUE)
plot_data_p_CL <- plot_data_p$fit - qt(0.975, length(Job[, 1])) *
    plot_data_p$se.fit
plot_data_p_CU <- plot_data_p$fit + qt(0.975, length(Job[, 1])) *
    plot_data_p$se.fit
plot_data_l <- predict(Job_BLR_2, predict_data, type = "link", se.fit = TRUE)
plot_data_l_CL <- plot_data_l$fit - qt(0.975, length(Job[, 1])) *
    plot_data_l$se.fit
plot_data_l_CU <- plot_data_l$fit + qt(0.975, length(Job[, 1])) *
    plot_data_l$se.fit
plot_data_o <- plot_data_p$fit/(1 - plot_data_p$fit)
plot_data_o_CL <- plot_data_p_CL/(1 - plot_data_p_CL)
plot_data_o_CU <- plot_data_p_CU/(1 - plot_data_p_CU)
plot_data <- as.data.frame(cbind(predict_data, plot_data_p$fit, plot_data_p_CL,
    plot_data_p_CU, plot_data_l$fit, plot_data_l_CL, plot_data_l_CU,
    plot_data_o, plot_data_o_CL, plot_data_o_CU))
names(plot_data) <- c("IV", "P", "P_CL", "P_CU", "L", "L_CL", "L_CU",
    "O", "O_CL", "O_CU")
plot_data$IV_Original <- seq(1000, 1600, 1)

ggplot(plot_data, aes(x = IV_Original, y = L)) + geom_line(size = 1,
    color = "blue") + geom_ribbon(data = plot_data, aes(ymin = L_CL,
    ymax = L_CU), alpha = 0.25) + coord_cartesian(xlim = c(1000, 1600),
    ylim = c(-8, 4)) + scale_x_continuous(breaks = c(seq(1000, 1600,
    100))) + scale_y_continuous(breaks = seq(-8, 4, 1)) + xlab("GRE") +
    ylab("Logit") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Logit as a \nFunction of GRE (95% CI)")
```



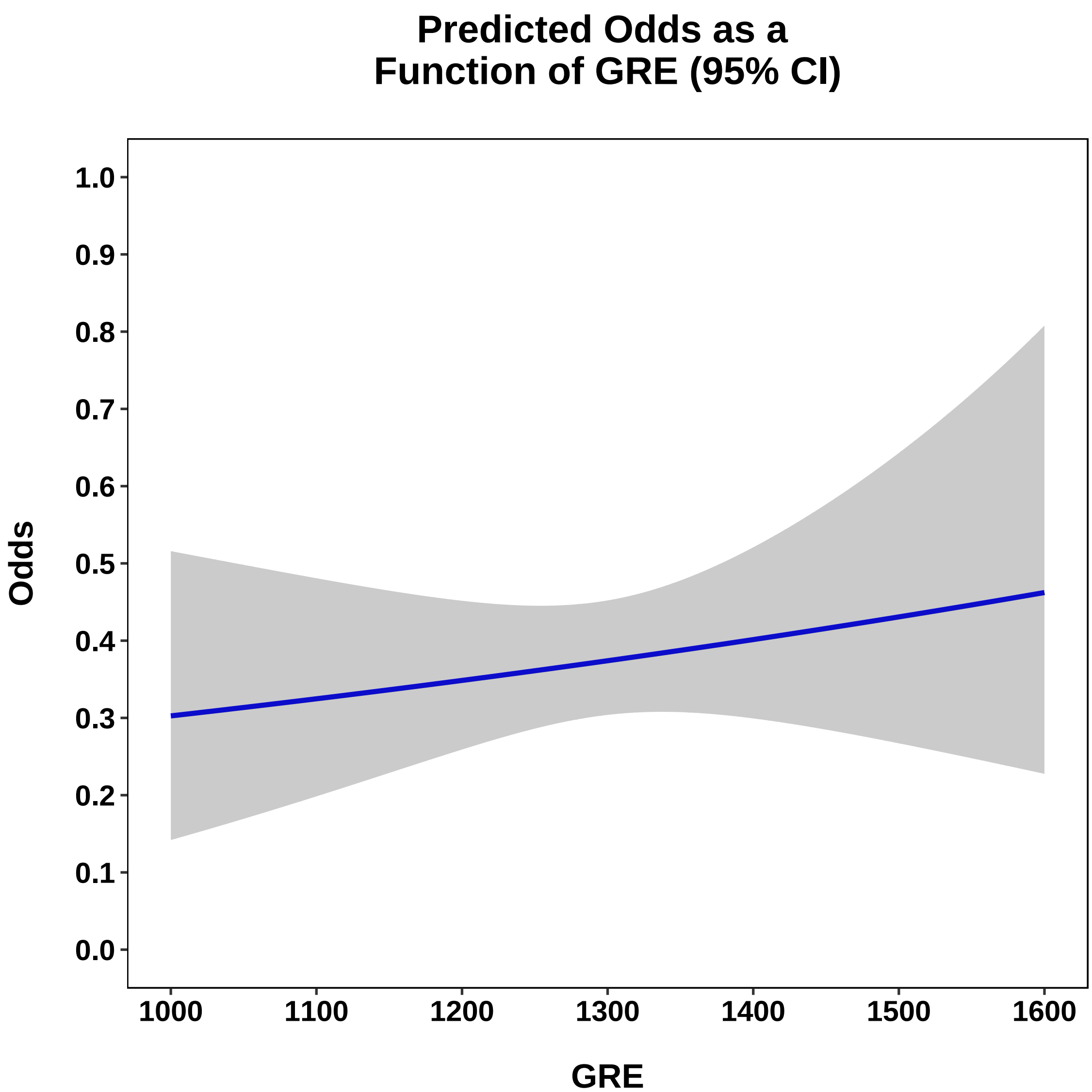

```
ggplot(plot_data, aes(x = IV_Original, y = P)) + geom_line(size = 1,
    color = "blue") + geom_ribbon(data = plot_data, aes(ymin = P_CL,
    ymax = P_CU), alpha = 0.25) + coord_cartesian(xlim = c(1000, 1600),
    ylim = c(0, 1)) + scale_x_continuous(breaks = c(seq(1000, 1600,
    100))) + scale_y_continuous(breaks = seq(0, 1, 0.1)) + xlab("GRE") +
    ylab("Probability") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
```

```
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Probability as a \nFunction of GRE (95% CI")
```




```
ggplot(plot_data, aes(x = IV_Original, y = O)) + geom_line(size = 1,
    color = "blue") + geom_ribbon(data = plot_data, aes(ymin = O_CL,
    ymax = O_CU), alpha = 0.25) + coord_cartesian(xlim = c(1000, 1600),
    ylim = c(0, 1)) + scale_x_continuous(breaks = c(seq(1000, 1600,
    100))) + scale_y_continuous(breaks = seq(0, 10, 0.1)) + xlab("GRE") +
    ylab("Odds") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
```

```
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Odds as a \nFunction of GRE (95% CI)")
```




```
Job_BLR_3 <- glm(job ~ pubs_c, family = binomial("logit"), data = Job)
summary(Job_BLR_3)

##
## Call:
```

```
## glm(formula = job ~ pubs_c, family = binomial("logit"), data = Job)
##
## Deviance Residuals:
##    Min       1Q  Median      3Q     Max
## -1.955  -0.493  -0.254   0.361   2.467
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.7585     0.1745   -10.1   <2e-16
## pubs_c        0.9491     0.0911    10.4   <2e-16
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 585.24  on 499  degrees of freedom
## Residual deviance: 367.62  on 498  degrees of freedom
## AIC: 371.6
##
## Number of Fisher Scoring iterations: 6
```

```
confint(Job_BLR_3)
```

```
## Waiting for profiling to be done...
```

```
##               2.5 % 97.5 %
## (Intercept) -2.1192 -1.433
## pubs_c       0.7802  1.138
```

```
confint.default(Job_BLR_3)
```

```
##               2.5 % 97.5 %
## (Intercept) -2.1006 -1.416
## pubs_c       0.7705  1.128
```

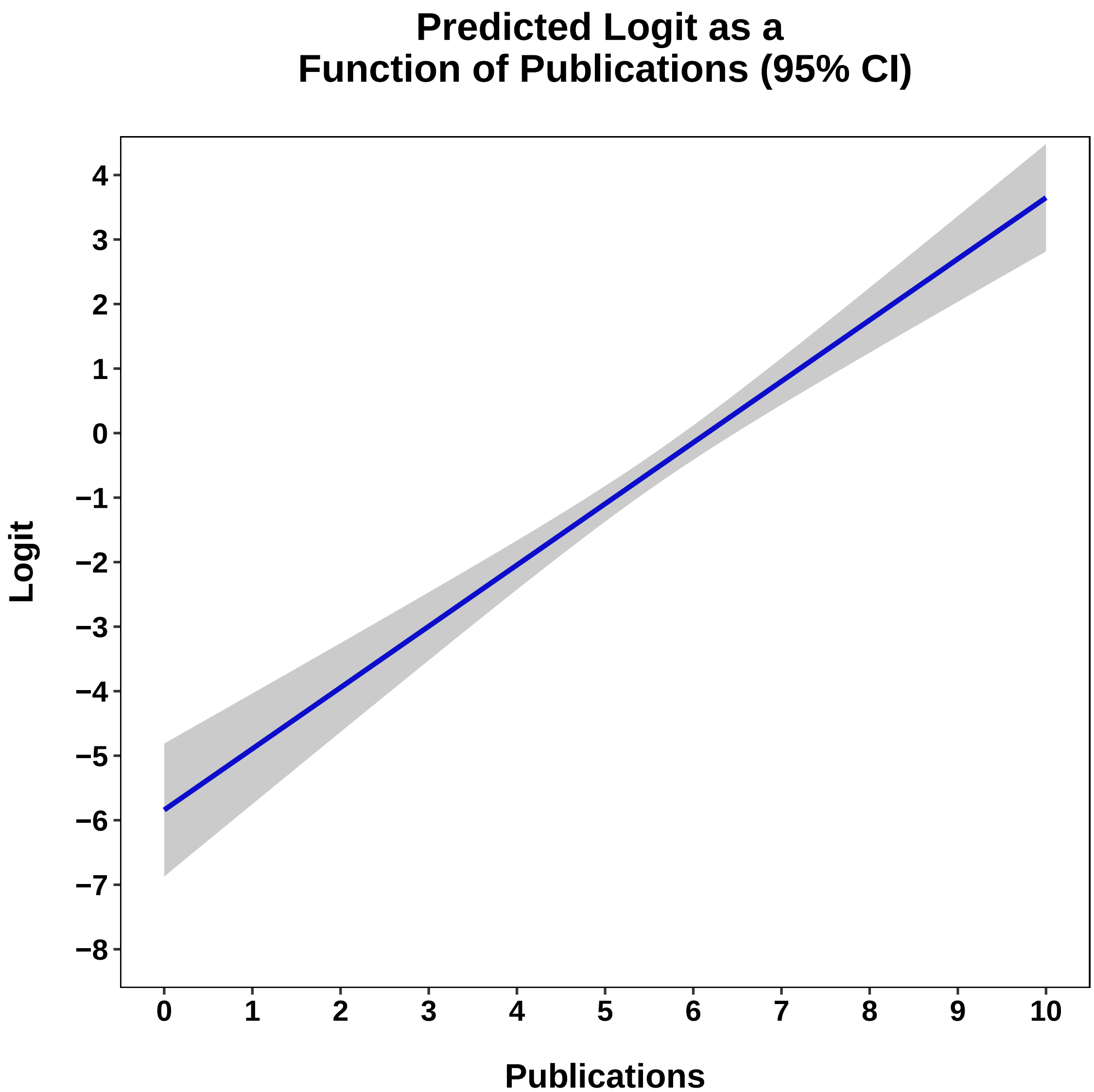
```
exp(cbind(OR = coef(Job_BLR_3), confint(Job_BLR_3)))
```

```
## Waiting for profiling to be done...
```

```
##                 OR  2.5 % 97.5 %
## (Intercept) 0.1723 0.1201 0.2386
## pubs_c      2.5833 2.1819 3.1212
```

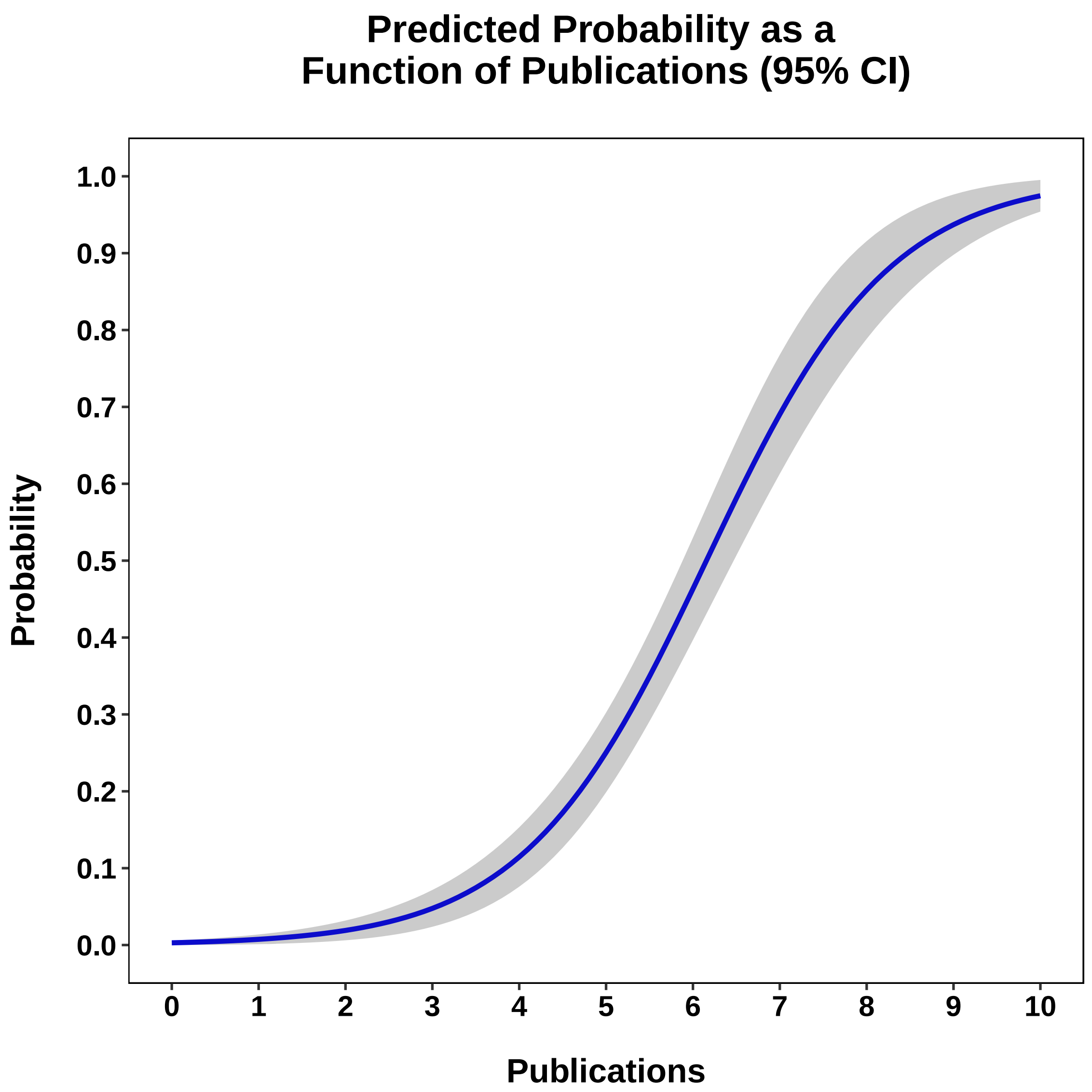
```
predict_data = with(Job, data.frame(pubs_c = seq(0 - mean(Job$pubs),
    10 - mean(Job$pubs), 0.1)))
plot_data_p <- predict(Job_BLR_3, predict_data, type = "response",
    se.fit = TRUE)
plot_data_p_CL <- plot_data_p$fit - qt(0.975, length(Job[, 1])) *
    plot_data_p$se.fit
plot_data_p_CU <- plot_data_p$fit + qt(0.975, length(Job[, 1])) *
    plot_data_p$se.fit
plot_data_l <- predict(Job_BLR_3, predict_data, type = "link", se.fit = TRUE)
plot_data_l_CL <- plot_data_l$fit - qt(0.975, length(Job[, 1])) *
    plot_data_l$se.fit
plot_data_l_CU <- plot_data_l$fit + qt(0.975, length(Job[, 1])) *
    plot_data_l$se.fit
plot_data_o <- plot_data_p$fit/(1 - plot_data_p$fit)
```

```
plot_data_o_CL <- plot_data_p_CL/(1 - plot_data_p_CL)
plot_data_o_CU <- plot_data_p_CU/(1 - plot_data_p_CU)
plot_data <- as.data.frame(cbind(predict_data, plot_data_p$fit, plot_data_p_CL,
    plot_data_p_CU, plot_data_l$fit, plot_data_l_CL, plot_data_l_CU,
    plot_data_o, plot_data_o_CL, plot_data_o_CU))
names(plot_data) <- c("IV", "P", "P_CL", "P_CU", "L", "L_CL", "L_CU",
    "O", "O_CL", "O_CU")
plot_data$IV_Original <- seq(0, 10, 0.1)

ggplot(plot_data, aes(x = IV_Original, y = L)) + geom_line(size = 1,
    color = "blue") + geom_ribbon(data = plot_data, aes(ymin = L_CL,
    ymax = L_CU), alpha = 0.25) + coord_cartesian(xlim = c(0, 10),
    ylim = c(-8, 4)) + scale_x_continuous(breaks = c(seq(0, 10, 1))) +
    scale_y_continuous(breaks = seq(-8, 4, 1)) + xlab("Publications") +
    ylab("Logit") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Logit as a \nFunction of Publications (95% CI)"
```



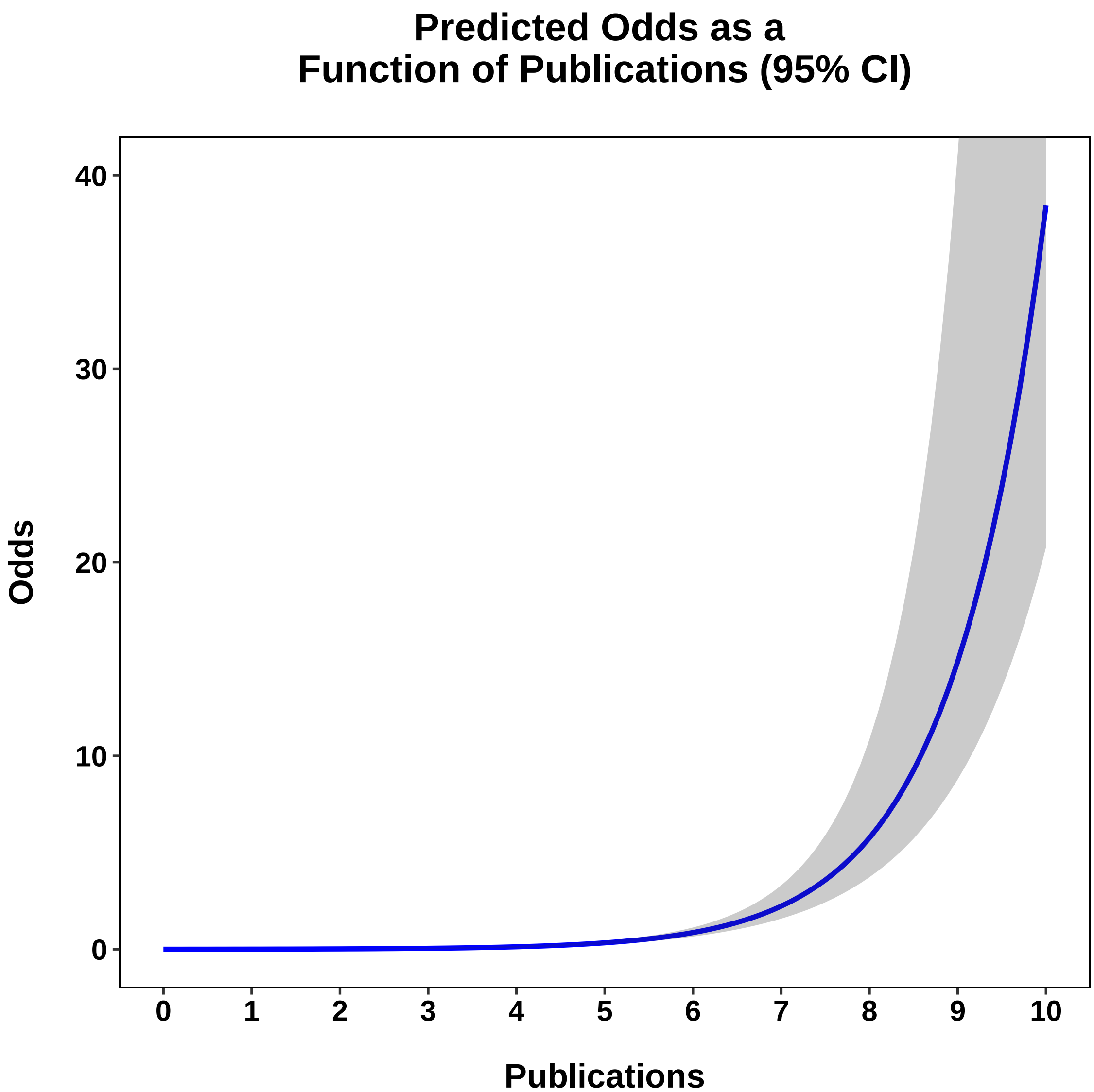

```
ggplot(plot_data, aes(x = IV_Original, y = P)) + geom_line(size = 1,
    color = "blue") + geom_ribbon(data = plot_data, aes(ymin = P_CL,
    ymax = P_CU), alpha = 0.25) + coord_cartesian(xlim = c(0, 10),
    ylim = c(0, 1)) + scale_x_continuous(breaks = c(seq(0, 10, 1))) +
    scale_y_continuous(breaks = seq(0, 1, 0.1)) + xlab("Publications") +
    ylab("Probability") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
```

```
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Probability as a \nFunction of Publications (95
```



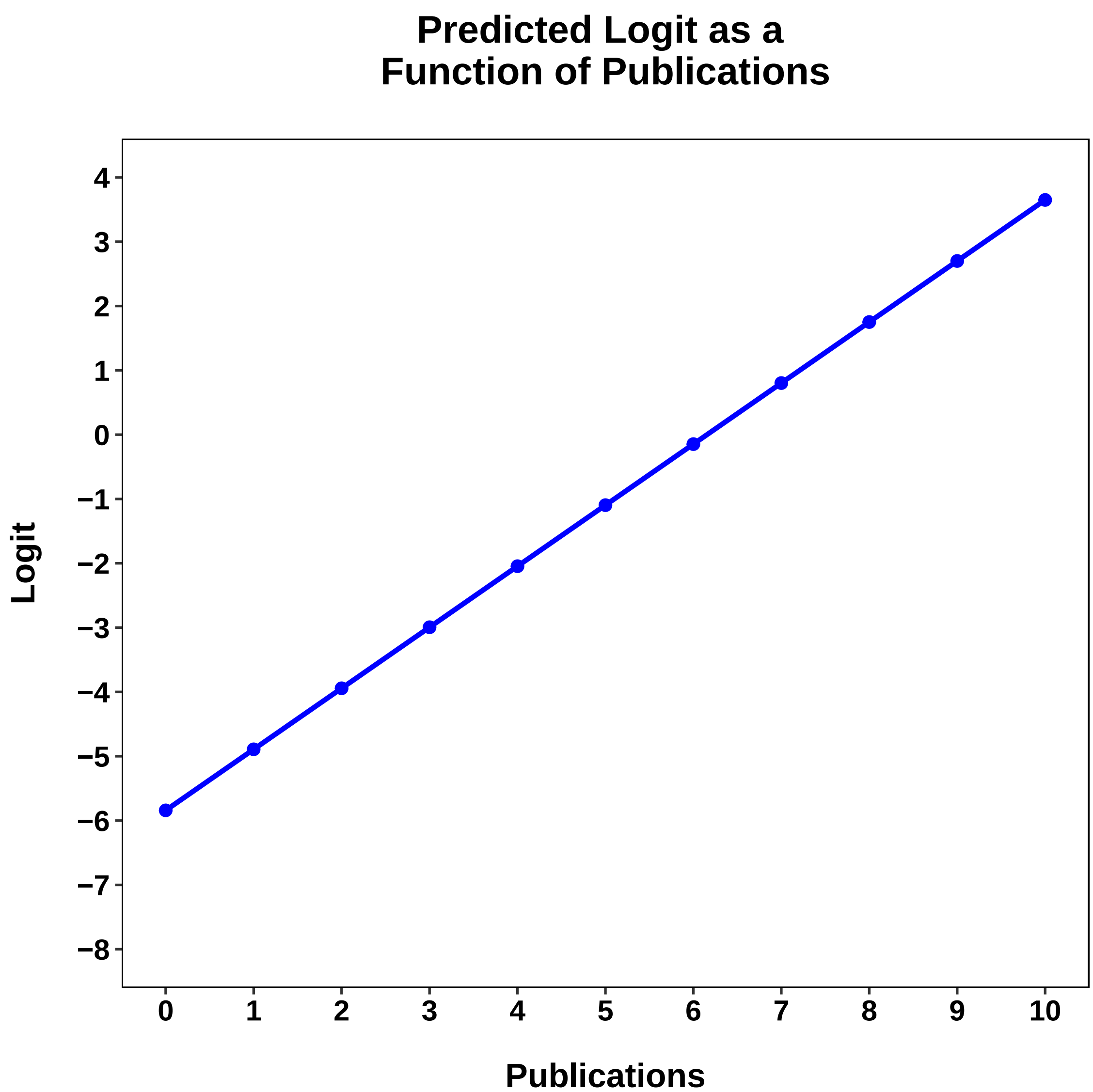

```
ggplot(plot_data, aes(x = IV_Original, y = O)) + geom_line(size = 1,
    color = "blue") + geom_ribbon(data = plot_data, aes(ymin = O_CL,
    ymax = O_CU), alpha = 0.25) + coord_cartesian(xlim = c(0, 10),
    ylim = c(0, 40)) + scale_x_continuous(breaks = c(seq(0, 10, 1))) +
    scale_y_continuous(breaks = seq(0, 40, 10)) + xlab("Publications") +
    ylab("Odds") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
```

```
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Odds as a \nFunction of Publications (95% CI)")
```



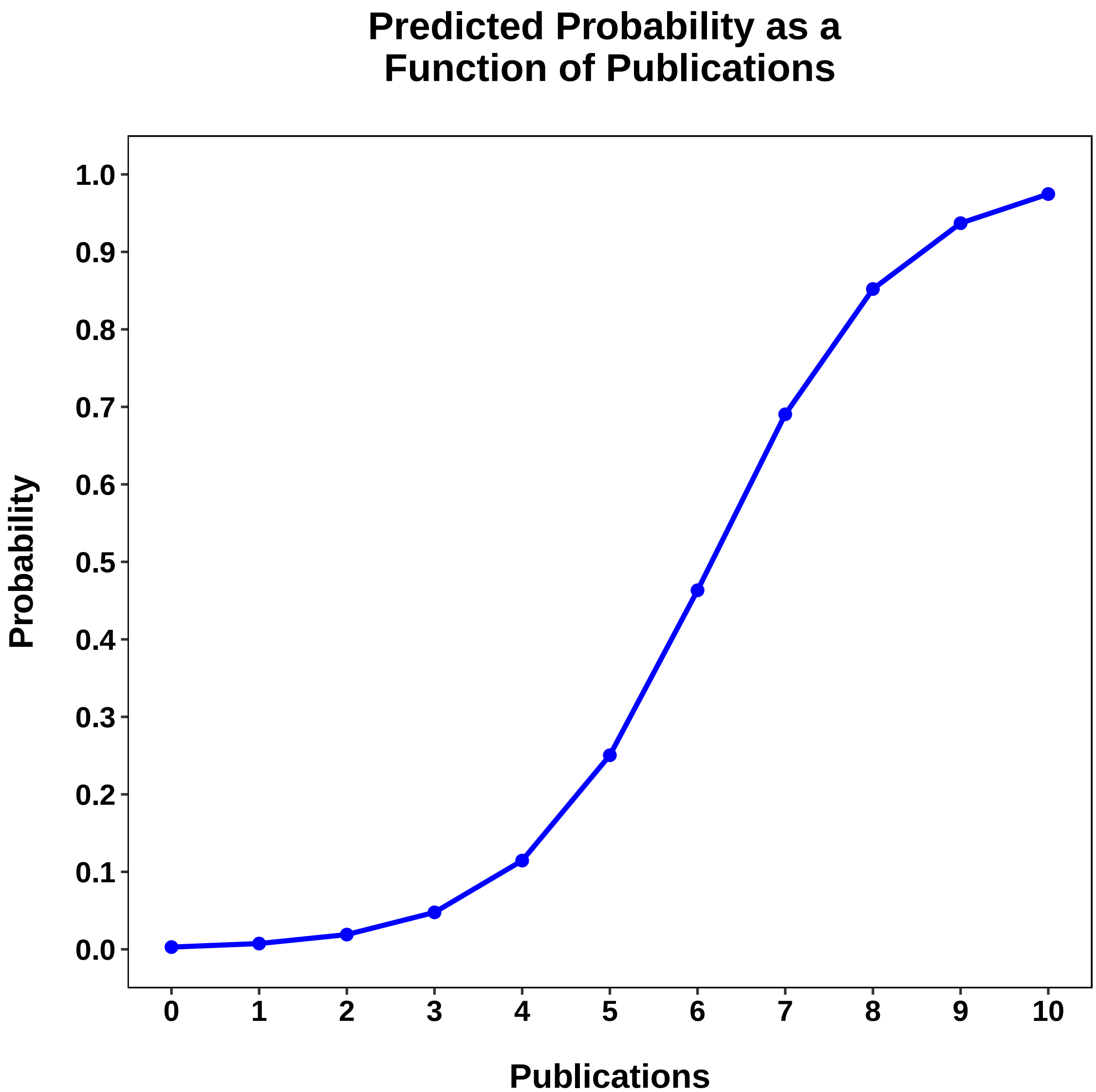

```
predict_data = with(Job, data.frame(pubs_c = seq(0 - mean(Job$pubs),
    10 - mean(Job$pubs), 1)))
plot_data_p <- predict(Job_BLR_3, predict_data, type = "response",
    se.fit = TRUE)
```

```r
plot_data_p_CL <- plot_data_p$fit - qt(0.975, length(Job[, 1])) *
    plot_data_p$se.fit
plot_data_p_CU <- plot_data_p$fit + qt(0.975, length(Job[, 1])) *
    plot_data_p$se.fit
plot_data_l <- predict(Job_BLR_3, predict_data, type = "link", se.fit = TRUE)
plot_data_l_CL <- plot_data_l$fit - qt(0.975, length(Job[, 1])) *
    plot_data_l$se.fit
plot_data_l_CU <- plot_data_l$fit + qt(0.975, length(Job[, 1])) *
    plot_data_l$se.fit
plot_data_o <- plot_data_p$fit/(1 - plot_data_p$fit)
plot_data_o_CL <- plot_data_p_CL/(1 - plot_data_p_CL)
plot_data_o_CU <- plot_data_p_CU/(1 - plot_data_p_CU)
plot_data <- as.data.frame(cbind(predict_data, plot_data_p$fit, plot_data_p_CL,
    plot_data_p_CU, plot_data_l$fit, plot_data_l_CL, plot_data_l_CU,
    plot_data_o, plot_data_o_CL, plot_data_o_CU))
names(plot_data) <- c("IV", "P", "P_CL", "P_CU", "L", "L_CL", "L_CU",
    "O", "O_CL", "O_CU")
plot_data$IV_Original <- seq(0, 10, 1)

ggplot(plot_data, aes(x = IV_Original, y = L)) + geom_point(size = 2,
    color = "blue") + geom_line(size = 1, color = "blue") + coord_cartesian(xlim = c(0,
    10), ylim = c(-8, 4)) + scale_x_continuous(breaks = c(seq(0, 10,
    1))) + scale_y_continuous(breaks = seq(-8, 4, 1)) + xlab("Publications") +
    ylab("Logit") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Logit as a \nFunction of Publications")
```



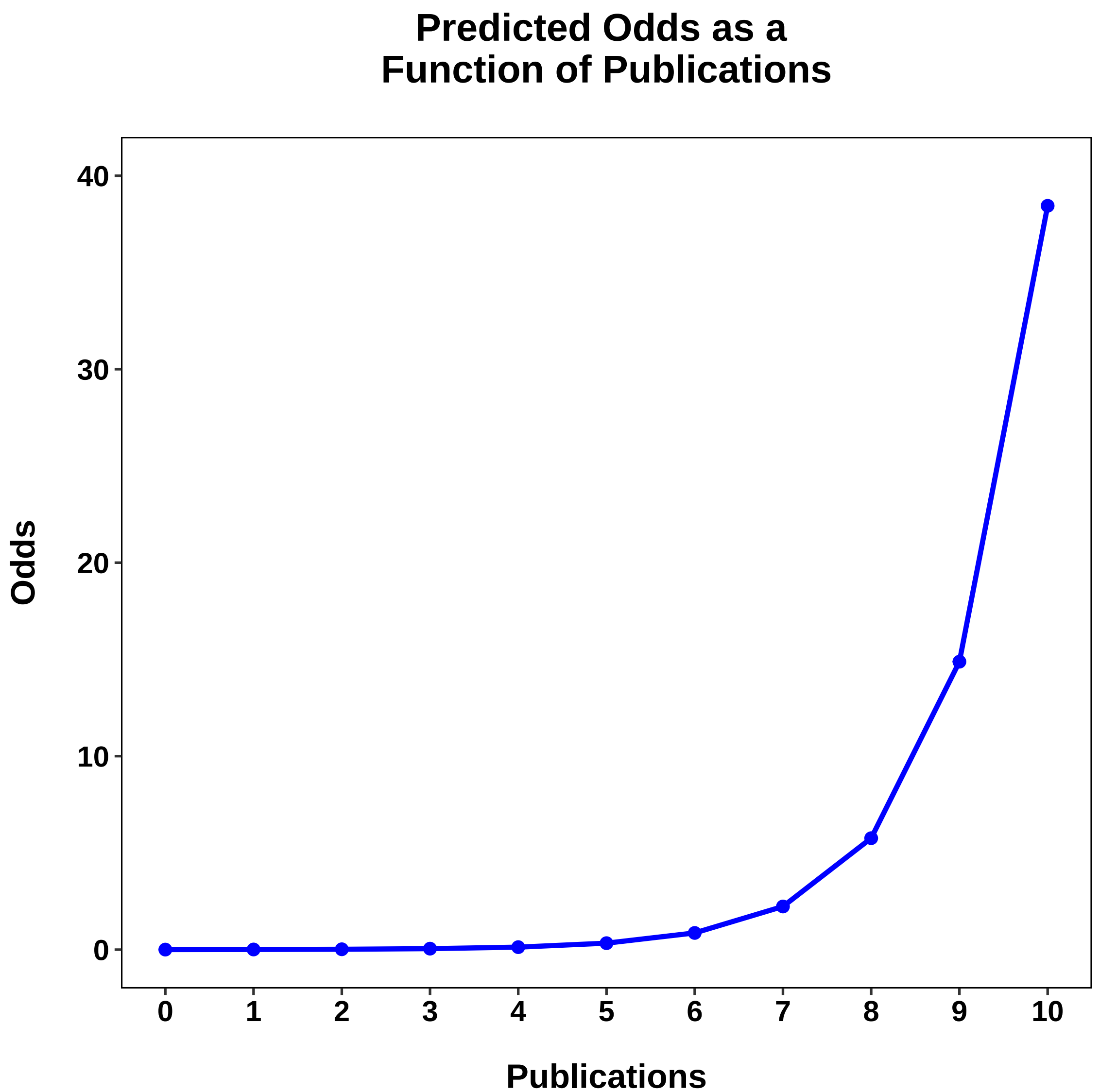

```
ggplot(plot_data, aes(x = IV_Original, y = P)) + geom_point(size = 2,
    color = "blue") + geom_line(size = 1, color = "blue") + coord_cartesian(xlim = c(0,
    10), ylim = c(0, 1)) + scale_x_continuous(breaks = c(seq(0, 10,
    1))) + scale_y_continuous(breaks = seq(0, 1, 0.1)) + xlab("Publications") +
    ylab("Probability") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
```

```
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Probability as a \nFunction of Publications")
```




```
ggplot(plot_data, aes(x = IV_Original, y = O)) + geom_point(size = 2,
    color = "blue") + geom_line(size = 1, color = "blue") + coord_cartesian(xlim = c(0,
    10), ylim = c(0, 40)) + scale_x_continuous(breaks = c(seq(0, 10,
    1))) + scale_y_continuous(breaks = seq(0, 40, 10)) + xlab("Publications") +
    ylab("Odds") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
```

```
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Odds as a \nFunction of Publications")
```




```
Job_BLR_4 <- glm(job ~ years_c, family = binomial("logit"), data = Job)
summary(Job_BLR_4)

##
## Call:
## glm(formula = job ~ years_c, family = binomial("logit"), data = Job)
##
```

```
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.142  -0.915  -0.554   1.213   2.446
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.2775     0.1307   -9.77  < 2e-16
## years_c      -0.5711     0.0786   -7.27  3.6e-13
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 585.24  on 499  degrees of freedom
## Residual deviance: 507.73  on 498  degrees of freedom
## AIC: 511.7
##
## Number of Fisher Scoring iterations: 5
```

```
confint(Job_BLR_4)
```

```
## Waiting for profiling to be done...
```

```
##               2.5 %  97.5 %
## (Intercept) -1.5452 -1.0310
## years_c     -0.7327 -0.4242
```

```
confint.default(Job_BLR_4)
```

```
##              2.5 %  97.5 %
## (Intercept) -1.534 -1.0212
## years_c     -0.725 -0.4171
```

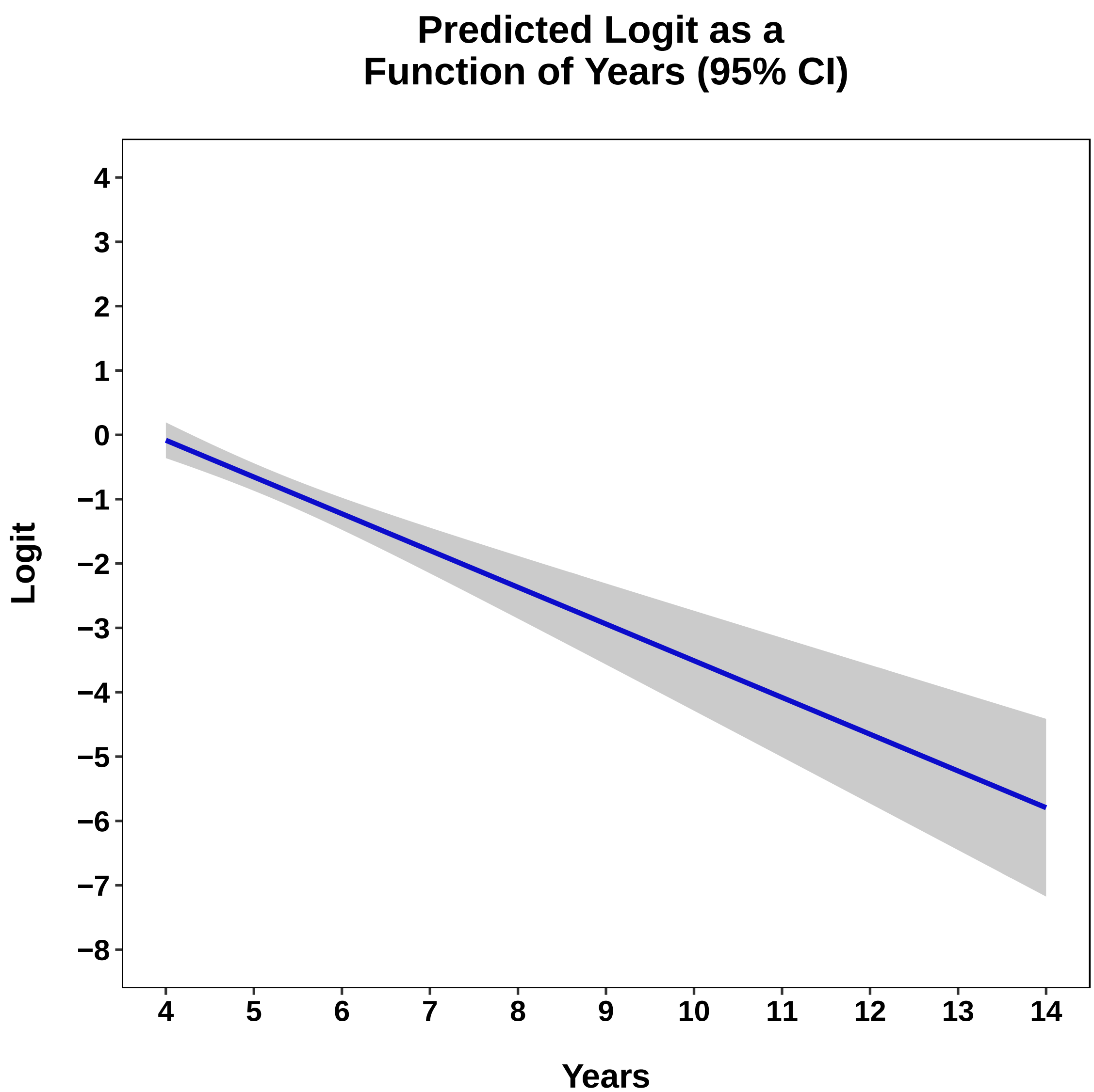
```
exp(cbind(OR = coef(Job_BLR_4), confint(Job_BLR_4)))
```

```
## Waiting for profiling to be done...
```

```
##                 OR  2.5 % 97.5 %
## (Intercept) 0.2787 0.2133 0.3566
## years_c     0.5649 0.4806 0.6543
```

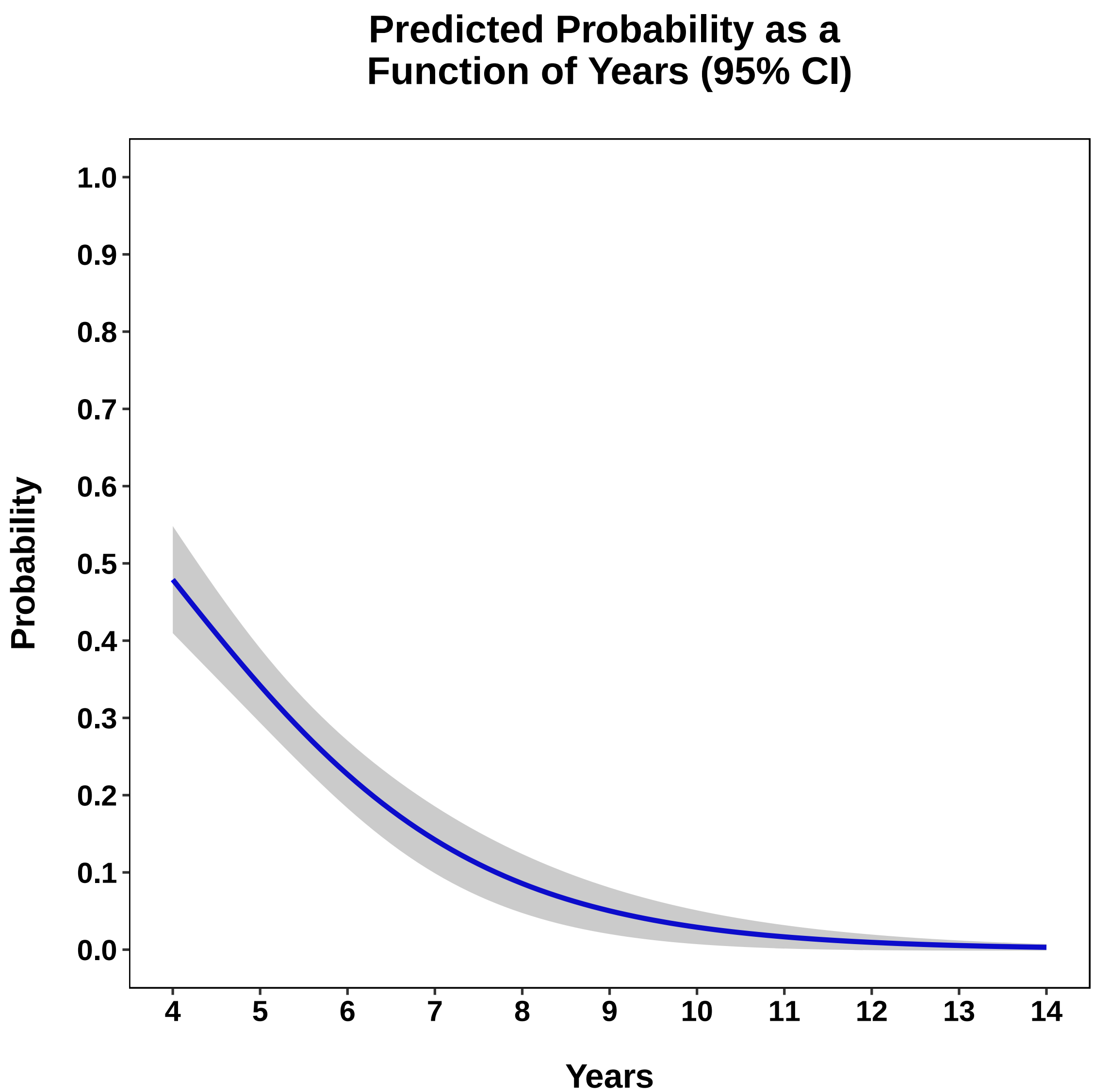
```
predict_data = with(Job, data.frame(years_c = seq(4 - mean(Job$years),
    14 - mean(Job$years), 0.1)))
plot_data_p <- predict(Job_BLR_4, predict_data, type = "response",
    se.fit = TRUE)
plot_data_p_CL <- plot_data_p$fit - qt(0.975, length(Job[, 1])) *
    plot_data_p$se.fit
plot_data_p_CU <- plot_data_p$fit + qt(0.975, length(Job[, 1])) *
    plot_data_p$se.fit
plot_data_l <- predict(Job_BLR_4, predict_data, type = "link", se.fit = TRUE)
plot_data_l_CL <- plot_data_l$fit - qt(0.975, length(Job[, 1])) *
    plot_data_l$se.fit
plot_data_l_CU <- plot_data_l$fit + qt(0.975, length(Job[, 1])) *
    plot_data_l$se.fit
plot_data_o <- plot_data_p$fit/(1 - plot_data_p$fit)
plot_data_o_CL <- plot_data_p_CL/(1 - plot_data_p_CL)
plot_data_o_CU <- plot_data_p_CU/(1 - plot_data_p_CU)
```

```r
plot_data <- as.data.frame(cbind(predict_data, plot_data_p$fit, plot_data_p_CL,
    plot_data_p_CU, plot_data_l$fit, plot_data_l_CL, plot_data_l_CU,
    plot_data_o, plot_data_o_CL, plot_data_o_CU))
names(plot_data) <- c("IV", "P", "P_CL", "P_CU", "L", "L_CL", "L_CU",
    "O", "O_CL", "O_CU")
plot_data$IV_Original <- seq(4, 14, 0.1)

ggplot(plot_data, aes(x = IV_Original, y = L)) + geom_line(size = 1,
    color = "blue") + geom_ribbon(data = plot_data, aes(ymin = L_CL,
    ymax = L_CU), alpha = 0.25) + coord_cartesian(xlim = c(4, 14),
    ylim = c(-8, 4)) + scale_x_continuous(breaks = c(seq(4, 14, 1))) +
    scale_y_continuous(breaks = seq(-8, 4, 1)) + xlab("Years") + ylab("Logit") +
    theme(text = element_text(size = 14, family = "sans", color = "black",
        face = "bold"), axis.text.y = element_text(colour = "black",
        size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
        0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
        15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold", margin = margin(0,
            0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
        legend.title = element_blank()) + ggtitle("Predicted Logit as a \nFunction of Years (95% CI)")
```



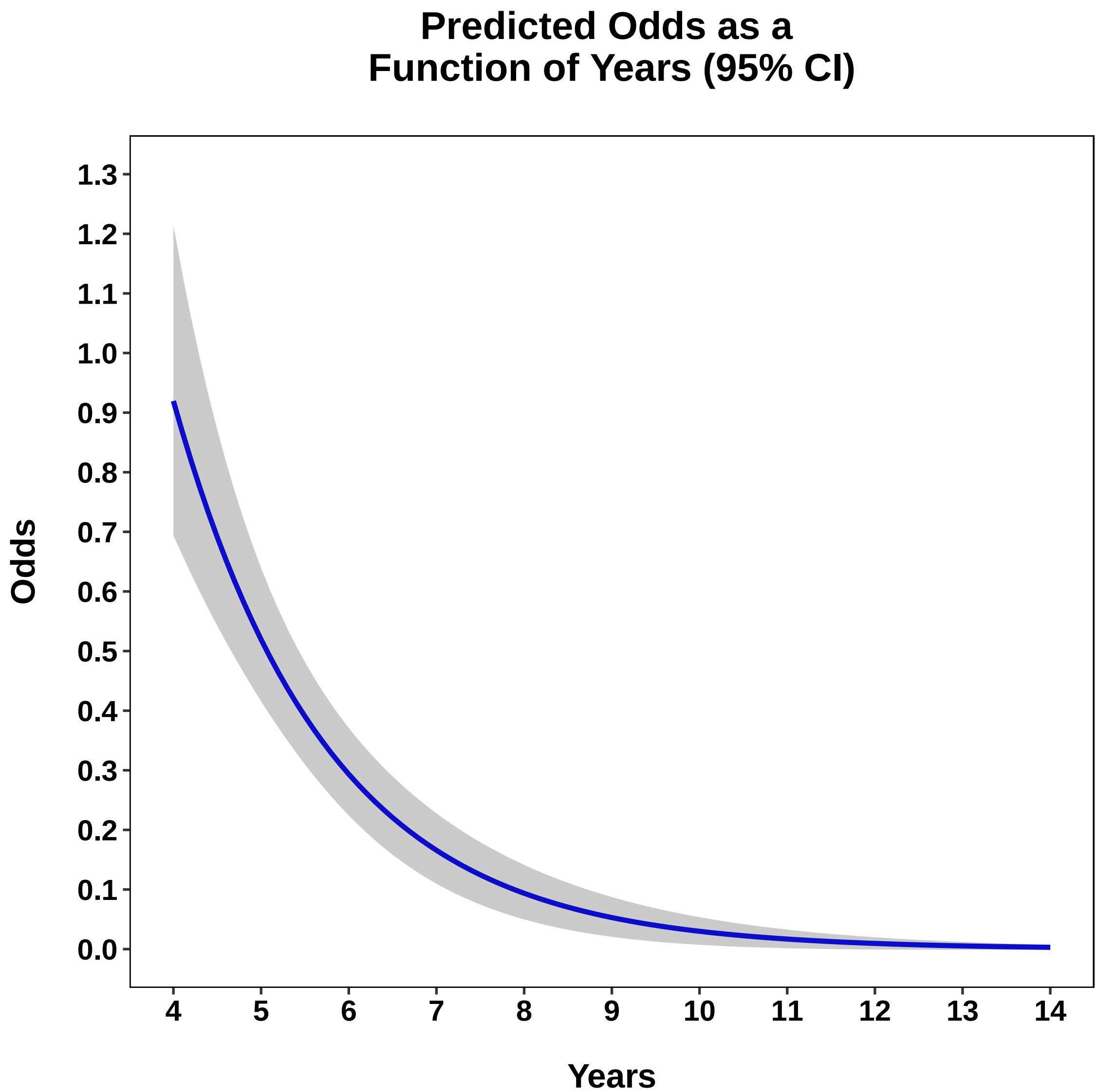

```
ggplot(plot_data, aes(x = IV_Original, y = P)) + geom_line(size = 1,
    color = "blue") + geom_ribbon(data = plot_data, aes(ymin = P_CL,
    ymax = P_CU), alpha = 0.25) + coord_cartesian(xlim = c(4, 14),
    ylim = c(0, 1)) + scale_x_continuous(breaks = c(seq(4, 14, 1))) +
    scale_y_continuous(breaks = seq(0, 1, 0.1)) + xlab("Years") +
    ylab("Probability") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
```

```
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Probability as a \nFunction of Years (95% CI)")
```




```
ggplot(plot_data, aes(x = IV_Original, y = O)) + geom_line(size = 1,
    color = "blue") + geom_ribbon(data = plot_data, aes(ymin = O_CL,
    ymax = O_CU), alpha = 0.25) + coord_cartesian(xlim = c(4, 14),
    ylim = c(0, 1.3)) + scale_x_continuous(breaks = c(seq(4, 14, 1))) +
    scale_y_continuous(breaks = seq(0, 1.3, 0.1)) + xlab("Years") +
    ylab("Odds") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
```

```
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Odds as a \nFunction of Years (95% CI)")
```




### 6.2 Multiple Predictor Models

```r
Job_BLR_5 <- glm(job ~ gre_c + pubs_c + years_c + sex_D, family = binomial("logit"),
    data = Job)
summary(Job_BLR_5)
```

```
##
## Call:
## glm(formula = job ~ gre_c + pubs_c + years_c + sex_D, family = binomial("logit"),
##     data = Job)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5596  -0.3111  -0.0142   0.0755   2.7257
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.71205    0.46554   -7.97  1.5e-15
## gre_c       -0.01470    0.00231   -6.37  1.8e-10
## pubs_c       1.99614    0.22058    9.05  < 2e-16
## years_c     -1.43390    0.18667   -7.68  1.6e-14
## sex_D       -0.40619    0.35023   -1.16     0.25
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 585.24  on 499  degrees of freedom
## Residual deviance: 225.23  on 495  degrees of freedom
## AIC: 235.2
##
## Number of Fisher Scoring iterations: 8
```

```r
confint(Job_BLR_5)
```

```
## Waiting for profiling to be done...
```

```
##               2.5 %   97.5 %
## (Intercept) -4.7000 -2.86634
## gre_c       -0.0195 -0.01042
## pubs_c       1.5989  2.46713
## years_c     -1.8281 -1.09360
## sex_D       -1.0989  0.27998
```

```r
confint.default(Job_BLR_5)
```

```
##                2.5 %   97.5 %
## (Intercept) -4.62450 -2.79960
## gre_c       -0.01922 -0.01018
## pubs_c       1.56380  2.42848
## years_c     -1.79977 -1.06803
## sex_D       -1.09262  0.28025
```

```r
exp(cbind(OR = coef(Job_BLR_5), confint(Job_BLR_5)))
```

```
## Waiting for profiling to be done...
```

```
##                  OR    2.5 %   97.5 %
## (Intercept) 0.02443 0.009095  0.05691
## gre_c       0.98540 0.980689  0.98963
```

```
## pubs_c      7.36059 4.947572 11.78854
## years_c     0.23838 0.160716  0.33501
## sex_D       0.66619 0.333227  1.32310

anova(Job_BLR_5)

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: job
##
## Terms added sequentially (first to last)
##
##
##         Df Deviance Resid. Df Resid. Dev
## NULL                      499        585
## gre_c    1      0.5       498        585
## pubs_c   1    239.3       497        345
## years_c  1    118.9       496        227
## sex_D    1      1.3       495        225

anova(Job_BLR_1, Job_BLR_5, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: job ~ sex_D
## Model 2: job ~ gre_c + pubs_c + years_c + sex_D
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       498        580
## 2       495        225  3      355   <2e-16

anova(Job_BLR_2, Job_BLR_5, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: job ~ gre_c
## Model 2: job ~ gre_c + pubs_c + years_c + sex_D
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       498        585
## 2       495        225  3      359   <2e-16

anova(Job_BLR_3, Job_BLR_5, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: job ~ pubs_c
## Model 2: job ~ gre_c + pubs_c + years_c + sex_D
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       498        368
## 2       495        225  3      142   <2e-16

anova(Job_BLR_4, Job_BLR_5, test = "Chisq")

## Analysis of Deviance Table
##
```

```
## Model 1: job ~ years_c
## Model 2: job ~ gre_c + pubs_c + years_c + sex_D
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       498        508
## 2       495        225  3      282   <2e-16
```

```r
wald.test(b = coef(Job_BLR_5), Sigma = vcov(Job_BLR_5), Terms = 5)
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 1.3, df = 1, P(> X2) = 0.25
```

```r
wald.test(b = coef(Job_BLR_5), Sigma = vcov(Job_BLR_5), Terms = 2:5)
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 85.5, df = 4, P(> X2) = 0.0
```

```r
wald.test(b = coef(Job_BLR_5), Sigma = vcov(Job_BLR_5), Terms = c(2,
    5))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 41.9, df = 2, P(> X2) = 7.9e-10
```

```r
Job_BLR_6 <- glm(job ~ gre_c + pubs_c + years_c + sex_D + pubs_c:years_c +
    pubs_c:sex_D + years_c:sex_D, family = binomial("logit"), data = Job)
summary(Job_BLR_6)
```

```
##
## Call:
## glm(formula = job ~ gre_c + pubs_c + years_c + sex_D + pubs_c:years_c +
##     pubs_c:sex_D + years_c:sex_D, family = binomial("logit"),
##     data = Job)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3329  -0.2921  -0.0155   0.0688   2.8001
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.52816    0.60351   -5.85  5.0e-09
## gre_c          -0.01476    0.00233   -6.32  2.6e-10
## pubs_c          1.78635    0.28812    6.20  5.6e-10
## years_c        -1.43511    0.32062   -4.48  7.6e-06
## sex_D          -0.68367    0.68462   -1.00     0.32
## pubs_c:years_c -0.02938    0.11205   -0.26     0.79
## pubs_c:sex_D    0.31366    0.31912    0.98     0.33
```

```
## years_c:sex_D   0.06856    0.32151    0.21     0.83
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 585.24  on 499  degrees of freedom
## Residual deviance: 223.54  on 492  degrees of freedom
## AIC: 239.5
##
## Number of Fisher Scoring iterations: 8
```

```
confint(Job_BLR_6)
```

```
## Waiting for profiling to be done...
```

```
##                    2.5 %   97.5 %
## (Intercept)    -4.85022 -2.46724
## gre_c          -0.01962 -0.01043
## pubs_c          1.26968  2.40327
## years_c        -2.12040 -0.85600
## sex_D          -2.03560  0.68572
## pubs_c:years_c -0.25275  0.18687
## pubs_c:sex_D   -0.32326  0.94346
## years_c:sex_D  -0.55388  0.72222
```

```
confint.default(Job_BLR_6)
```

```
##                    2.5 %   97.5 %
## (Intercept)    -4.71102 -2.34529
## gre_c          -0.01934 -0.01018
## pubs_c          1.22164  2.35106
## years_c        -2.06351 -0.80671
## sex_D          -2.02549  0.65816
## pubs_c:years_c -0.24900  0.19023
## pubs_c:sex_D   -0.31179  0.93912
## years_c:sex_D  -0.56159  0.69872
```

```
exp(cbind(OR = coef(Job_BLR_6), confint(Job_BLR_6)))
```

```
## Waiting for profiling to be done...
```

```
##                      OR    2.5 %   97.5 %
## (Intercept)    0.02936 0.007827  0.08482
## gre_c          0.98535 0.980569  0.98963
## pubs_c         5.96764 3.559724 11.05927
## years_c        0.23809 0.119984  0.42486
## sex_D          0.50476 0.130603  1.98520
## pubs_c:years_c 0.97104 0.776663  1.20548
## pubs_c:sex_D   1.36843 0.723789  2.56885
## years_c:sex_D  1.07097 0.574714  2.05900
```

```
wald.test(b = coef(Job_BLR_6), Sigma = vcov(Job_BLR_6), Terms = 6:8)
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 1.6, df = 3, P(> X2) = 0.65
```

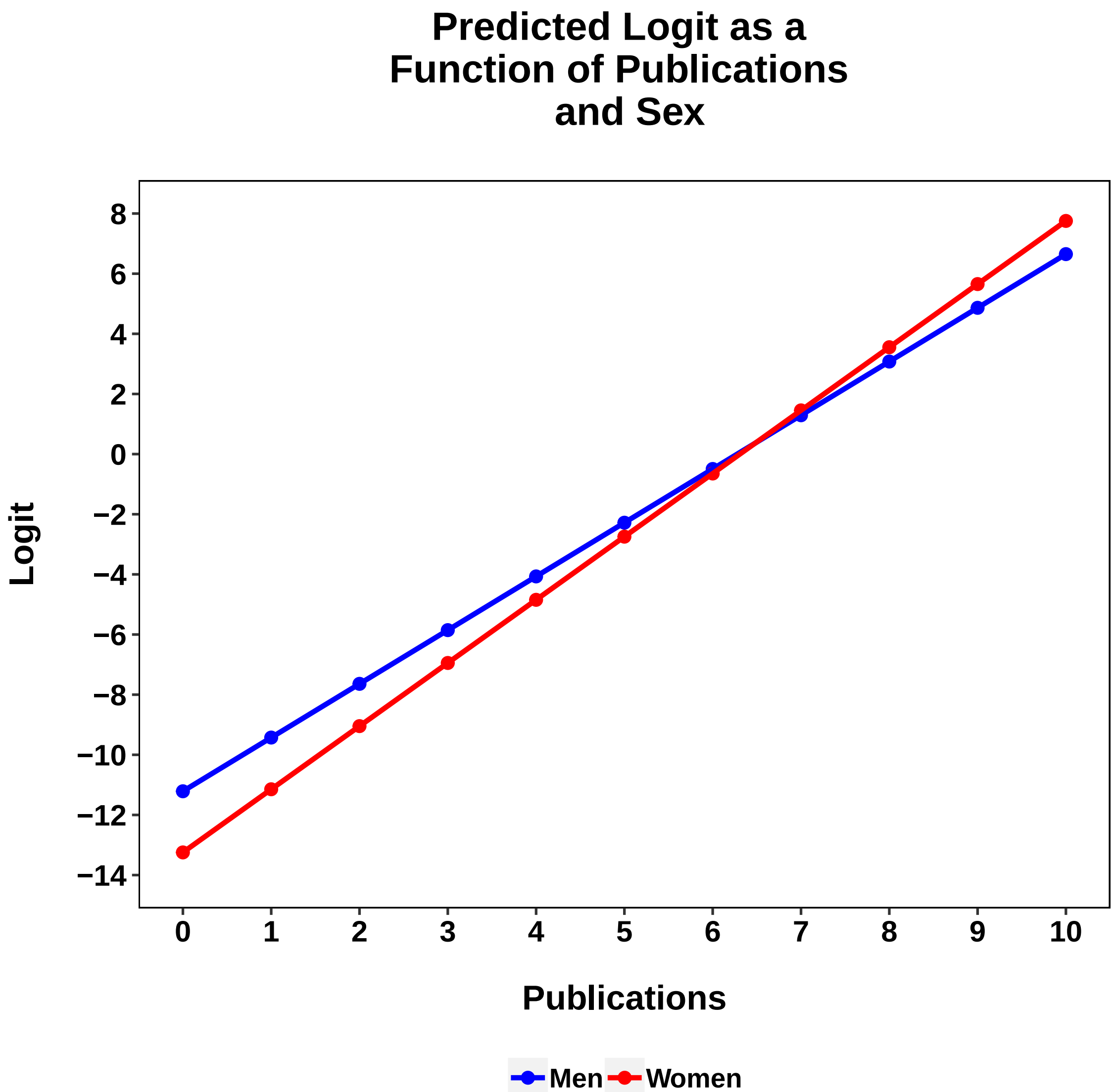
```r
# Reduce list of coefficients to separate models for men and women
# containing only a constant and a term for publications.
Men_Constant <- coef(Job_BLR_6)[1]
Men_Pubs <- coef(Job_BLR_6)[3]
Women_Constant <- coef(Job_BLR_6)[1] + coef(Job_BLR_6)[5]
Women_Pubs <- coef(Job_BLR_6)[3] + coef(Job_BLR_6)[7]

Men_Constant

## (Intercept)
##      -3.528

Men_Pubs

## pubs_c
##  1.786

Women_Constant

## (Intercept)
##      -4.212

Women_Pubs

## pubs_c
##    2.1

# Odds for men and women.
exp(Men_Pubs)

## pubs_c
##  5.968

exp(Women_Pubs)

## pubs_c
##  8.166

# Odds ratio.
exp(Women_Pubs)/exp(Men_Pubs)

## pubs_c
##  1.368
```

```r
Job_BLR_6_No_I <- glm(job ~ -1 + M_D + F_D + M_D:gre_c + M_D:pubs_c +
    M_D:years_c + M_D:pubs_c:years_c + F_D:gre_c + F_D:pubs_c + F_D:years_c +
    F_D:pubs_c:years_c, family = binomial("logit"), data = Job)
summary(Job_BLR_6_No_I)

##
## Call:
## glm(formula = job ~ -1 + M_D + F_D + M_D:gre_c + M_D:pubs_c +
##     M_D:years_c + M_D:pubs_c:years_c + F_D:gre_c + F_D:pubs_c +
##     F_D:years_c + F_D:pubs_c:years_c, family = binomial("logit"),
##     data = Job)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2924  -0.2948  -0.0144   0.0613   2.8598
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## M_D                -3.46388    0.72614   -4.77  1.8e-06
## F_D                -4.30736    0.76659   -5.62  1.9e-08
## M_D:gre_c          -0.01286    0.00342   -3.76  0.00017
## M_D:pubs_c          1.72613    0.34034    5.07  3.9e-07
## M_D:years_c        -1.43858    0.39932   -3.60  0.00032
## F_D:gre_c          -0.01623    0.00321   -5.05  4.4e-07
## F_D:pubs_c          2.17365    0.37028    5.87  4.4e-09
## F_D:years_c        -1.37644    0.39696   -3.47  0.00053
## M_D:pubs_c:years_c  0.01604    0.16412    0.10  0.92214
## F_D:pubs_c:years_c -0.06469    0.15084   -0.43  0.66801
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 693.15  on 500  degrees of freedom
## Residual deviance: 222.88  on 490  degrees of freedom
## AIC: 242.9
##
## Number of Fisher Scoring iterations: 9
```

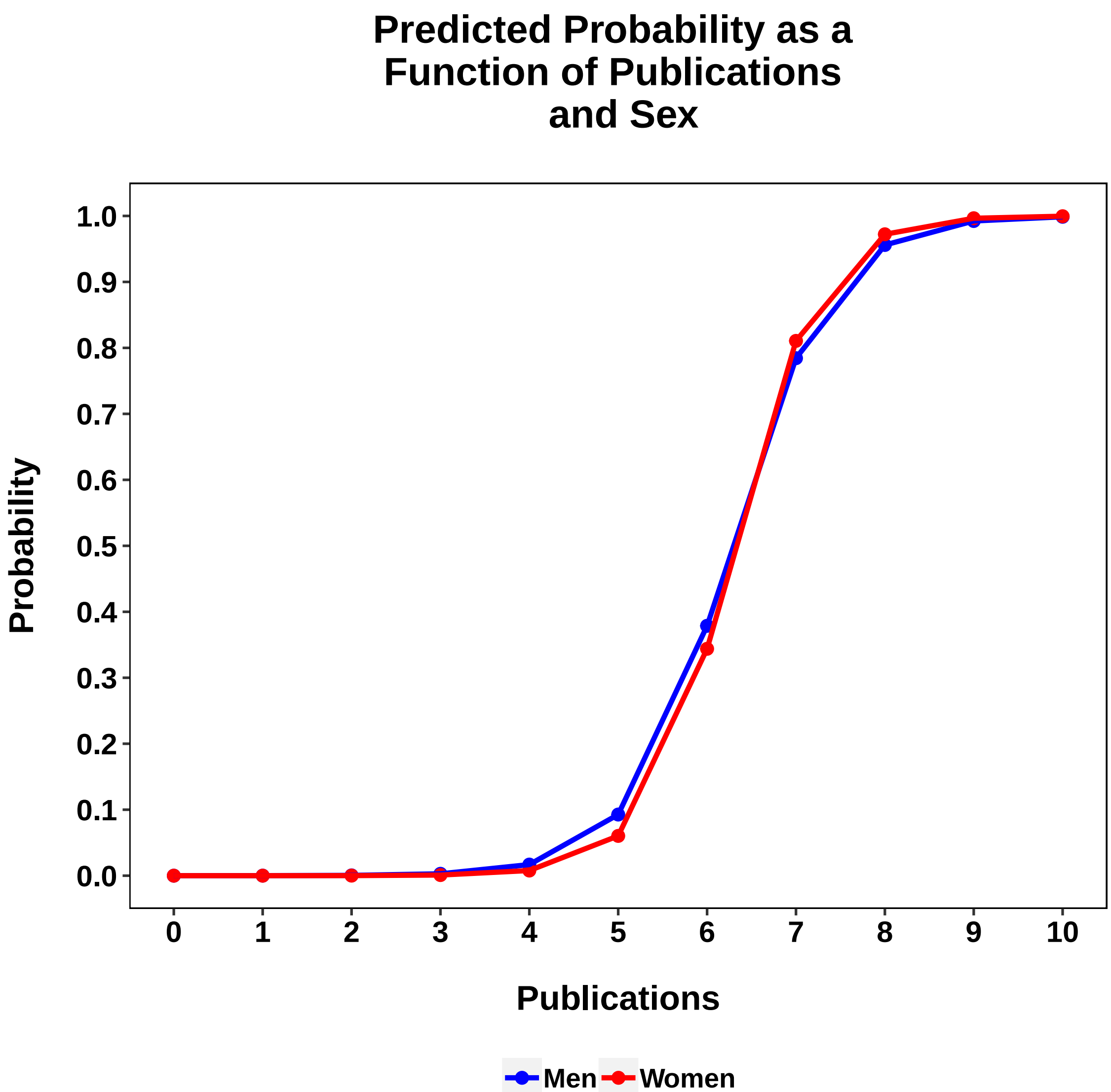
```
predict_data <- with(Job, data.frame(gre_c = mean(gre_c), years_c = mean(years_c),
    expand.grid(pubs_c = seq(from = 0 - mean(pubs), to = 10 - mean(pubs),
        by = 1), sex_D = c(0, 1))))
plot_data_P <- predict(Job_BLR_6, predict_data, type = "response")
plot_data_L <- predict(Job_BLR_6, predict_data, type = "link")
plot_data_O <- plot_data_P/(1 - plot_data_P)
plot_data <- as.data.frame(cbind(predict_data, plot_data_P, plot_data_L,
    plot_data_O))
names(plot_data) <- c("gre_mean", "years_mean", "pubs_c", "sex", "P",
    "L", "O")
plot_data$IV_Original <- rep(seq(0, 10, 1), 2)
plot_data$sex_F <- factor(plot_data$sex, levels = c(0, 1), labels = c("Men",
    "Women"))

ggplot(plot_data, aes(x = IV_Original, y = L, group = sex_F)) + geom_point(aes(color = sex_F),
    size = 2) + geom_line(aes(color = sex_F), size = 1) + scale_color_manual(values = c("blue",
    "red")) + coord_cartesian(xlim = c(0, 10), ylim = c(-14, 8)) +
    scale_x_continuous(breaks = c(seq(0, 10, 1))) + scale_y_continuous(breaks = seq(-14,
    8, 2)) + xlab("Publications") + ylab("Logit") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
```

```
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Logit as a \nFunction of Publications \n and Se
```



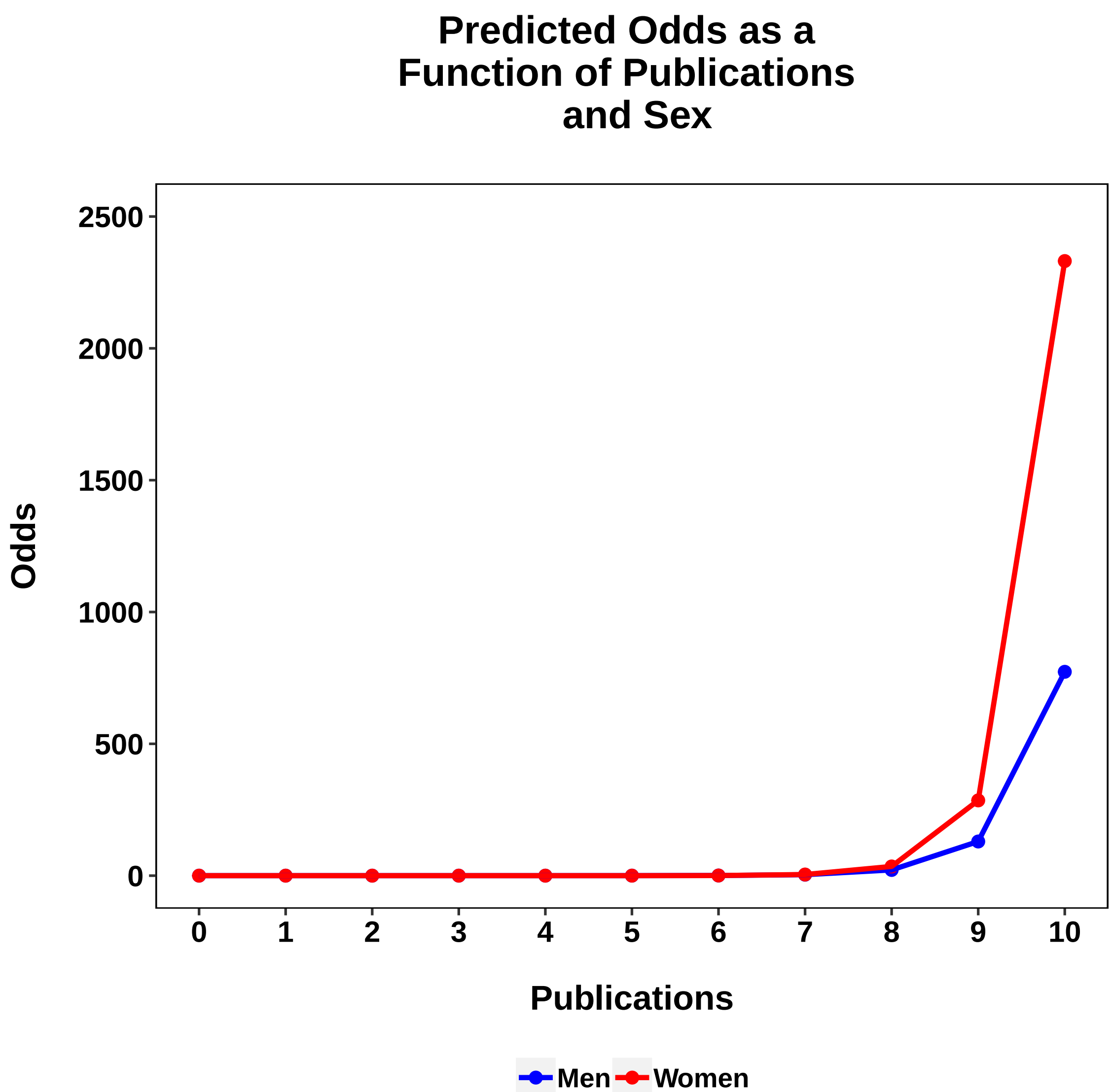

```
ggplot(plot_data, aes(x = IV_Original, y = P, group = sex_F)) + geom_point(aes(color = sex_F),
    size = 2) + geom_line(aes(color = sex_F), size = 1) + scale_color_manual(values = c("blue",
    "red")) + coord_cartesian(xlim = c(0, 10), ylim = c(0, 1)) + scale_x_continuous(breaks = c(seq(0,
    10, 1))) + scale_y_continuous(breaks = seq(0, 1, 0.1)) + xlab("Publications") +
    ylab("Probability") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
```

```
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Probability as a \nFunction of Publications \n
```




*Note that the following graph can be easily misinterpreted. The odds ratio for the interaction is NOT the ratio of the odds for men versus women in the graph for any given value of publications. Instead, it is the ratio of the odds ratio for men relative to the odds ratio for women. The ratio of consecutive values in the graph, for men and women separately, gives the odds ratios for men and women respectively (higher odds relative to lower odds).*

```r
ggplot(plot_data, aes(x = IV_Original, y = 0, group = sex_F)) + geom_point(aes(color = sex_F),
    size = 2) + geom_line(aes(color = sex_F), size = 1) + scale_color_manual(values = c("blue",
    "red")) + coord_cartesian(xlim = c(0, 10), ylim = c(0, 2500)) +
    scale_x_continuous(breaks = c(seq(0, 10, 1))) + scale_y_continuous(breaks = seq(0,
    2500, 500)) + xlab("Publications") + ylab("Odds") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Odds as a \nFunction of Publications \n and Sex
```




```r
Job_BLR_7 <- glm(Job$job ~ gre_c + pubs_c + years_c + sex_D + I(pubs_c^2),
    family = binomial("logit"), data = Job)
summary(Job_BLR_7)

##
## Call:
## glm(formula = Job$job ~ gre_c + pubs_c + years_c + sex_D + I(pubs_c^2),
##     family = binomial("logit"), data = Job)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4270  -0.2966  -0.0129   0.0954   2.7408
##
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.67453    0.46818   -7.85  4.2e-15
## gre_c       -0.01465    0.00229   -6.39  1.7e-10
## pubs_c       2.16589    0.30909    7.01  2.4e-12
## years_c     -1.41059    0.18569   -7.60  3.0e-14
## sex_D       -0.40424    0.35093   -1.15     0.25
## I(pubs_c^2) -0.07396    0.08421   -0.88     0.38
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 585.24  on 499  degrees of freedom
## Residual deviance: 224.44  on 494  degrees of freedom
## AIC: 236.4
##
## Number of Fisher Scoring iterations: 8
```

```
confint(Job_BLR_7)
```

```
## Waiting for profiling to be done...
```

```
##                2.5 %   97.5 %
## (Intercept) -4.66680 -2.82249
## gre_c       -0.01943 -0.01039
## pubs_c       1.62844  2.84131
## years_c     -1.80368 -1.07260
## sex_D       -1.09859  0.28308
## I(pubs_c^2) -0.24003  0.08591
```

```
confint.default(Job_BLR_7)
```

```
##                2.5 %   97.5 %
## (Intercept) -4.59214 -2.75692
## gre_c       -0.01915 -0.01016
## pubs_c       1.56008  2.77170
## years_c     -1.77453 -1.04664
## sex_D       -1.09205  0.28357
## I(pubs_c^2) -0.23900  0.09108
```

```
exp(cbind(OR = coef(Job_BLR_7), confint(Job_BLR_7)))
```

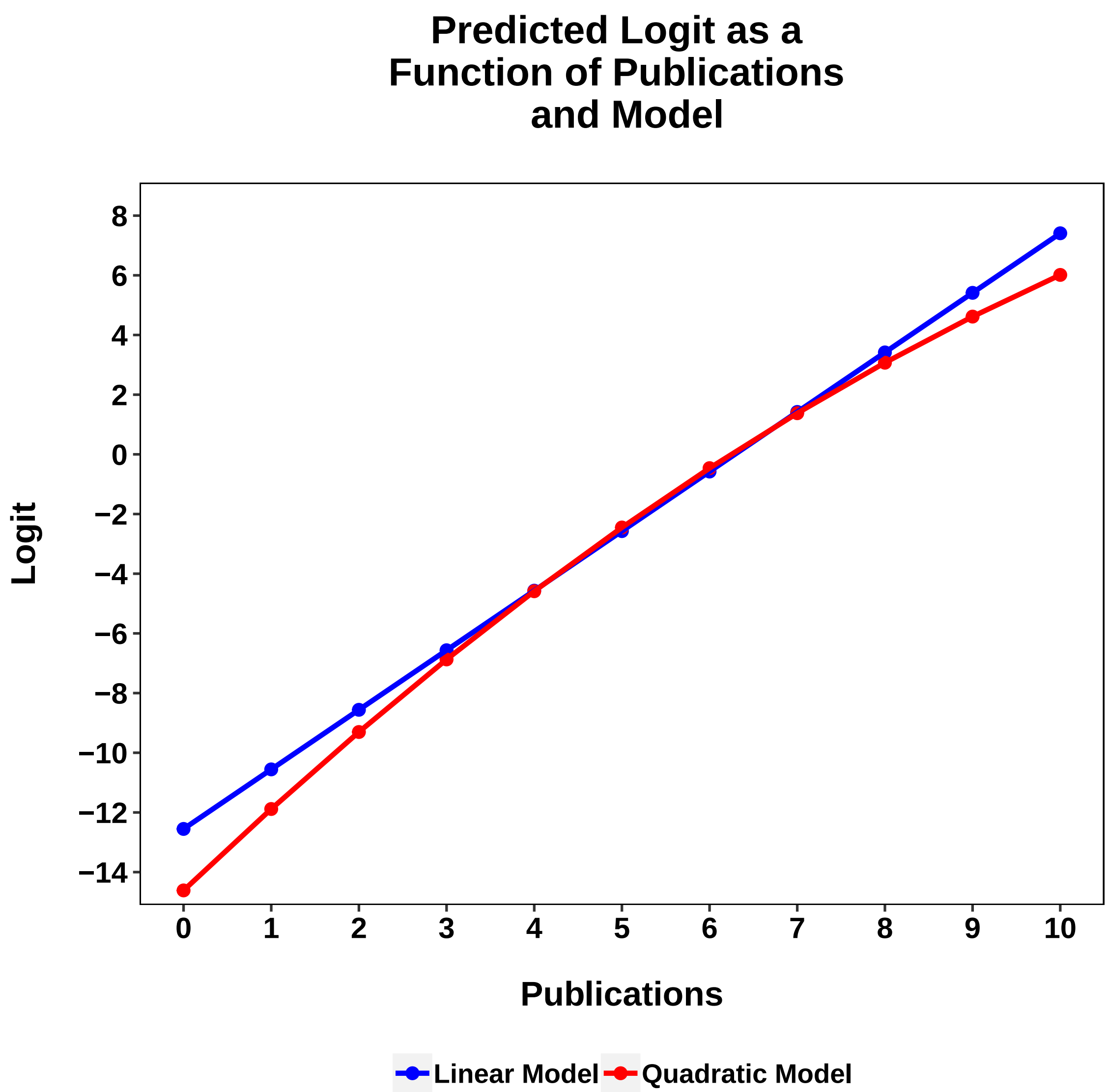
```
## Waiting for profiling to be done...
```

```
##                  OR    2.5 %   97.5 %
## (Intercept) 0.02536 0.009402  0.05946
## gre_c       0.98545 0.980762  0.98966
## pubs_c      8.72235 5.095901 17.13827
## years_c     0.24400 0.164692  0.34212
## sex_D       0.66748 0.333340  1.32721
## I(pubs_c^2) 0.92871 0.786603  1.08971
```

```
plot_data_linear <- with(Job, data.frame(gre_c = mean(gre_c), years_c = mean(years_c),
    sex_D = mean(sex_D), pubs_c = seq(from = 0 - mean(pubs), to = 10 -
        mean(pubs), by = 1)))
plot_data_linear$P <- predict(Job_BLR_5, newdata = plot_data_linear,
    type = "response")
```

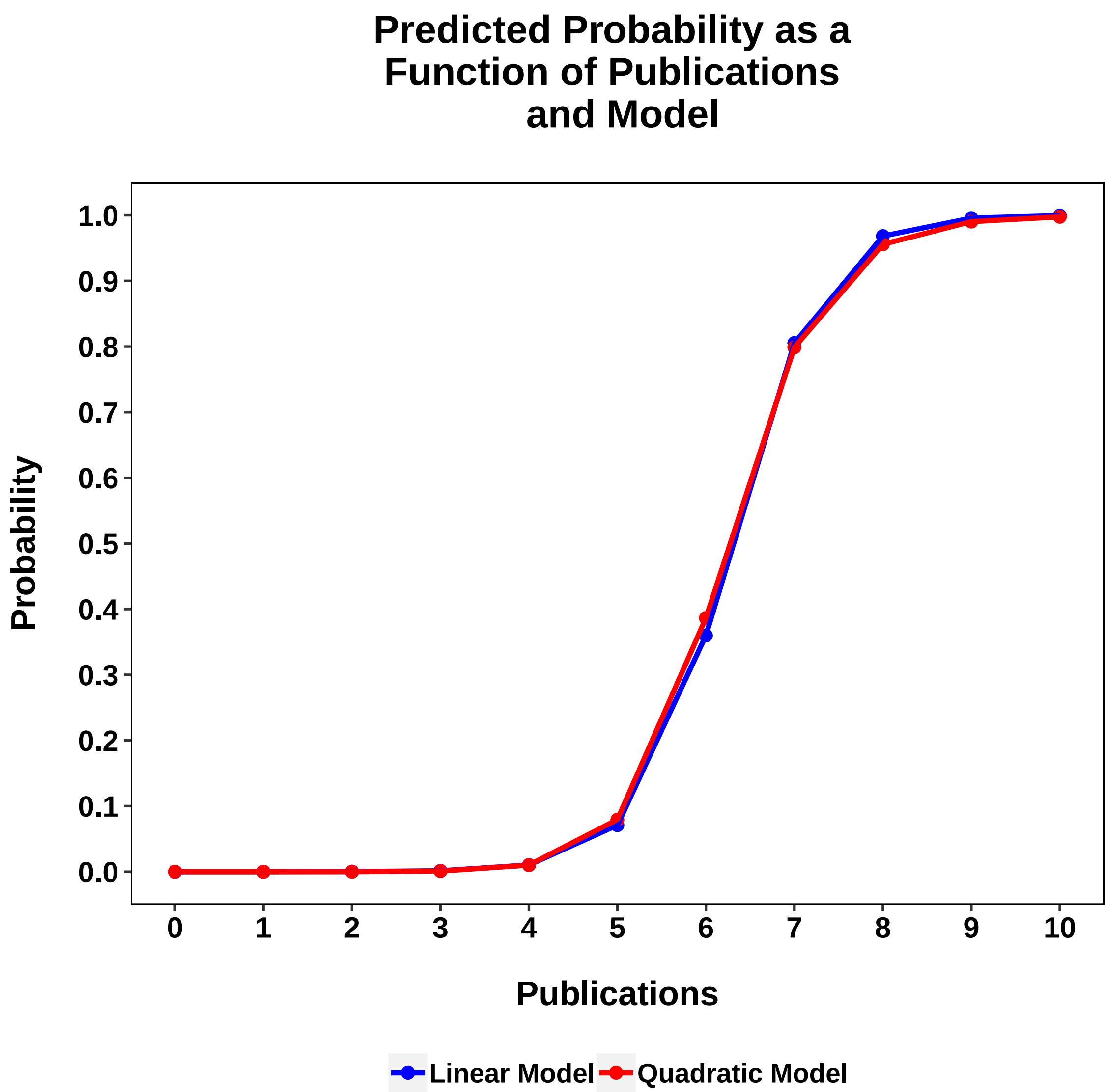
```r
plot_data_linear$L <- predict(Job_BLR_5, newdata = plot_data_linear)
plot_data_linear$O <- plot_data_linear$P/(1 - plot_data_linear$P)
plot_data_quad <- with(Job, data.frame(gre_c = mean(gre_c), years_c = mean(years_c),
    sex_D = mean(sex_D), pubs_c = seq(from = 0 - mean(pubs), to = 10 -
        mean(pubs), by = 1)))
plot_data_quad$P <- predict(Job_BLR_7, newdata = plot_data_quad, type = "response")
plot_data_quad$L <- predict(Job_BLR_7, newdata = plot_data_quad)
plot_data_quad$O <- plot_data_quad$P/(1 - plot_data_quad$P)
plot_data <- rbind(plot_data_linear, plot_data_quad)
plot_data$IV_Original <- rep(seq(0, 10, 1), 2)
plot_data$model <- c(rep("Linear", 11), rep("Quadratic", 11))
plot_data$model_F <- factor(plot_data$model, levels = c("Linear",
    "Quadratic"), labels = c("Linear Model", "Quadratic Model"))
```

```r
ggplot(plot_data, aes(x = IV_Original, y = L, group = model_F)) +
    geom_point(aes(color = model_F), size = 2) + geom_line(aes(color = model_F),
    size = 1) + scale_color_manual(values = c("blue", "red")) + coord_cartesian(xlim = c(0,
    10), ylim = c(-14, 8)) + scale_x_continuous(breaks = c(seq(0,
    10, 1))) + scale_y_continuous(breaks = seq(-14, 8, 2)) + xlab("Publications") +
    ylab("Logit") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Logit as a \nFunction of Publications \n and Mo
```



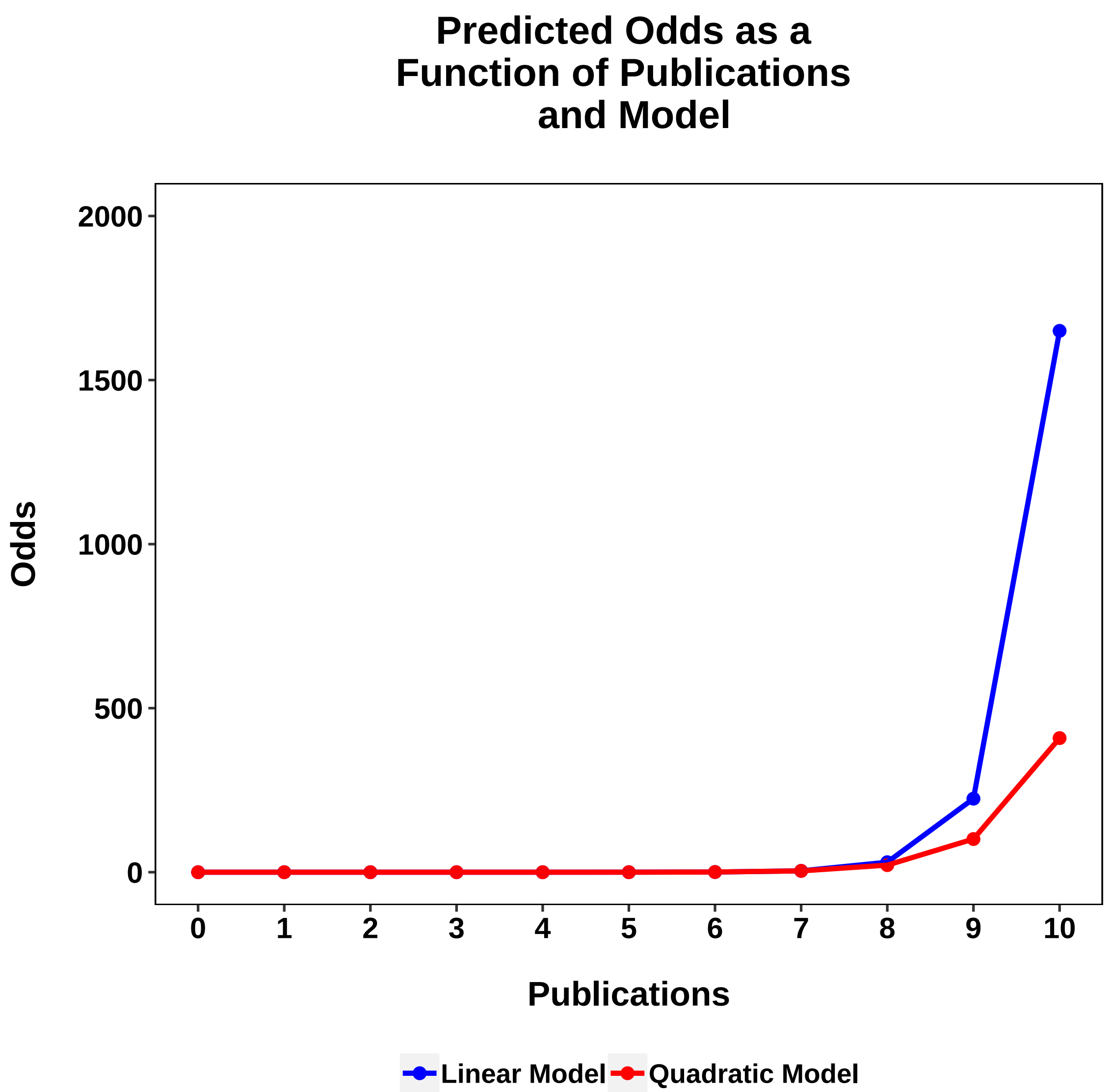

```
ggplot(plot_data, aes(x = IV_Original, y = P, group = model_F)) +
    geom_point(aes(color = model_F), size = 2) + geom_line(aes(color = model_F),
    size = 1) + scale_color_manual(values = c("blue", "red")) + coord_cartesian(xlim = c(0,
    10), ylim = c(0, 1)) + scale_x_continuous(breaks = c(seq(0, 10,
    1))) + scale_y_continuous(breaks = seq(0, 1, 0.1)) + xlab("Publications") +
    ylab("Probability") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
```

```
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Probability as a \nFunction of Publications \n
```




```
ggplot(plot_data, aes(x = IV_Original, y = 0, group = model_F)) +
    geom_point(aes(color = model_F), size = 2) + geom_line(aes(color = model_F),
    size = 1) + scale_color_manual(values = c("blue", "red")) + coord_cartesian(xlim = c(0,
    10), ylim = c(0, 2000)) + scale_x_continuous(breaks = c(seq(0,
    10, 1))) + scale_y_continuous(breaks = seq(0, 2000, 500)) + xlab("Publications") +
    ylab("Odds") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
```

```
    0, 0, 0), size = 14), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 14), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Predicted Odds as a \nFunction of Publications \n and Mod
```



### 6.3 Classification

***The predicted logit for each person can be transformed to a probability and then used to classify cases into predicted job status. These predictions can be compared to actual job status in a manner that resembles that used in discriminant analysis. The same indices of classification quality can be applied. For example, Klecka'a ($\tau$) can be used:***

$$\tau = \frac{N_o - \sum_{i=1}^{G} p_i n_i}{N - \sum_{i=1}^{G} p_i n_i}$$

$N_o$ ***is the observed number of correct classifications,*** $n_i$ ***is the number of cases in group i,*** $p_i$ ***is the proportion of the total sample expected to be in group i, G is the number of groups, and N is the total sample size.***

***The confusionMatrix( ) function from the caret package provides quite a number of other indices. Of note, it provides Cohen's kappa, a chance-corrected agreement statistic. If the data are arranged in a confusion table as follows:***

| | | Actual | | |
|---|---|---|---|---|
| | | *Absent* | *Present* | *Marginal* |
| Prediction | Absent | d | c | Row 1 = d+c |
| | Present | b | a | Row 2 = b+a |
| | Marginal | Column 1 = d+b | Column 2 = c+a | N=a+b+c+d |

***Cohen's*** $\kappa$ ***is defined as:***

$$\kappa = \frac{N_o - N_e}{N - N_e} = \frac{p_o - p_e}{1 - p_e}$$

$N_o$ ***is the number of correct classifications (the sum of the main diagonal: d+a).*** $N_e$ ***is the number of correct classifications expected by chance. This is calculated using the marginals: [(Row 1 x Column 1)+(Row 2 x Column 2)]/N. N is the total sample size. Or,*** $\kappa$ ***can be estimated with proportions,*** $p_o$ ***and*** $p_e$***.***

***The other indices reported are likewise a function of elements in the table. They are commonly used when there are only two groups (as here). When there are more than two groups, then these are also reported, but for all combinations of each group versus the combination of the remaining groups. Some of the more useful are precision, recall, and F1.***

***Precision is defined as:***

$$Precision = \frac{a}{a+b}$$

***This index answers the question, "What percentage of predicted events are correct?"***

***Recall is defined as:***

$$Recall = \frac{a}{a+c}$$

***This index answers the question, "What percentage of events were correctly predicted?"***

***Precision and recall are negatively related; as one increases, the other decreases. An index that combines them is F1, defined as:***

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

***This is the harmonic mean of precision and recall.***

***Some of the remaining indices are often useful. These are the most common.***

***Sensitivity is defined as:***

$$Sensitivity = \frac{a}{a + c}$$

***Sensitivity is the same as recall: "What percentage of events were correctly predicted?"***

***Specificity is defined as:***

$$Specificity = \frac{d}{b + d}$$

***This index answers the question: "What percentage of event absences were correctly predicted?"***

***Prevalence is defined as:***

$$Prevalence = \frac{a + c}{a + b + c + d}$$

***This index aswers the question: "What is the proportion of actual events in the sample?"***

***Detection Rate is defined as:***

$$Detection\ \ Rate = \frac{a}{a + b + c + d}$$

***This index answers the question: "What proportion of the entire sample are correctly predicted events?"***

***Detection Prevalence is defined as:***

$$Detection\ \ Prevalence = \frac{a + b}{a + b + c + d}$$

***This index answers the question: "What proportion of the entire sample are predicted events?"***

```
Job$job_P <- ifelse(Job_BLR_5$fitted.values > 0.5, 1, 0)
Class_T <- table(Original = Job$job, Predicted = Job$job_P)
PC <- sum(diag(Class_T))/sum(Class_T)
MO1 <- sum(Class_T[1, ])
```

```
MO2 <- sum(Class_T[2, ])
MP1 <- MO1
MP2 <- MO2
N <- sum(Class_T)
O <- sum(diag(Class_T))
E <- (MO1 * MP1/N) + (MO2 * MP2/N)
Tau <- (O - E)/(N - E)
t <- (O - E)/sqrt(length(Job[, 1]) * (E/length(Job[, 1])) * (1 - E/length(Job[,
    1])))
T

## [1] TRUE

PC

## [1] 0.888

Tau

## [1] 0.7172

t

## [1] 12.99
```

```
chi_squared <- (((O - E)^2)/E) + ((((500 - O) - (500 - E))^2)/(500 -
    E))
chi_squared

## [1] 168.6
```

```
Job_Predicted <- factor(Job$job_P, levels = c(0, 1), labels = c("No Job",
    "Job"))
Job_Actual <- factor(Job$job, levels = c(0, 1), labels = c("No Job",
    "Job"))
confusionMatrix(Job_Predicted, Job_Actual, positive = "Job", mode = "everything")

## Confusion Matrix and Statistics
##
##             Reference
## Prediction No Job Job
##     No Job    339  31
##     Job        25 105
##
##                Accuracy : 0.888
##                  95% CI : (0.857, 0.914)
##     No Information Rate : 0.728
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.713
##  Mcnemar's Test P-Value : 0.504
##
##             Sensitivity : 0.772
##             Specificity : 0.931
```

```
##          Pos Pred Value : 0.808
##          Neg Pred Value : 0.916
##               Precision : 0.808
##                  Recall : 0.772
##                      F1 : 0.789
##              Prevalence : 0.272
##          Detection Rate : 0.210
##    Detection Prevalence : 0.260
##       Balanced Accuracy : 0.852
##
##        'Positive' Class : Job
##
```