Homework 2

Sarah Adilijiang

Problem 3

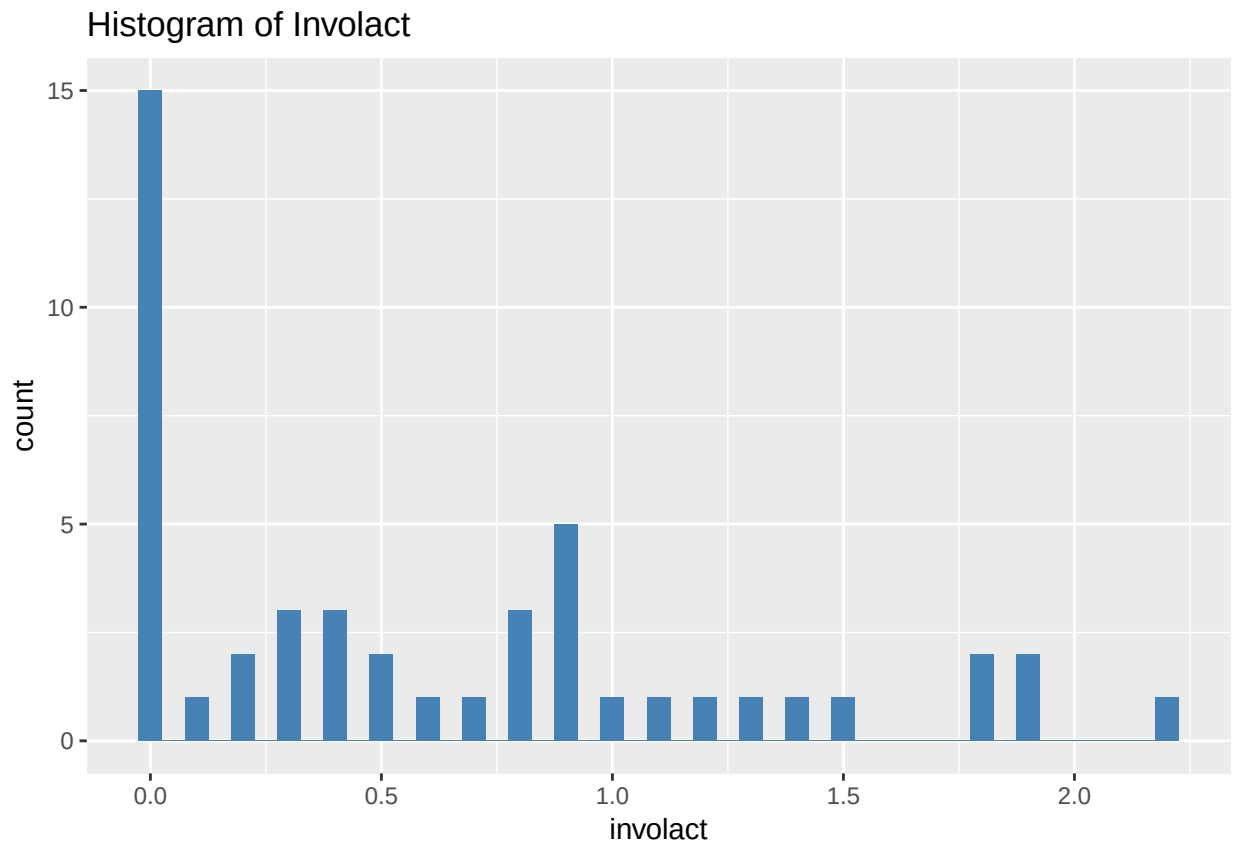
(a)

```
library(faraway)
data(chredlin)

# histogram of "involact"
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
ggplot(chredlin, aes(x=involact)) +
    geom_histogram(binwidth=0.05, fill="steelblue") +
    labs(title="Histogram of Involact")
```




```
# fraction of the zero responses
n = nrow(chredlin)
15/n
```

```
## [1] 0.3191489
```

Answer:

31.91% of the "involact" responses are zeros.

(b)

Gaussian model:

$involact_i = \beta_0 + \beta_1 race_i + \beta_2 fire_i + \beta_3 theft_i + \beta_4 age_i + \beta_5 log(income_i) + \epsilon_i$

```
# ignore "side" and fit the Gaussian model
str(chredlin)
```

```
## 'data.frame':    47 obs. of  7 variables:
##  $ race    : num  10 22.2 19.6 17.3 24.5 54 4.9 7.1 5.3 21.5 ...
##  $ fire    : num  6.2 9.5 10.5 7.7 8.6 34.1 11 6.9 7.3 15.1 ...
##  $ theft   : num  29 44 36 37 53 68 75 18 31 25 ...
##  $ age     : num  60.4 76.5 73.5 66.9 81.4 52.6 42.6 78.5 90.1 89.8 ...
##  $ involact: num  0 0.1 1.2 0.5 0.7 0.3 0 0 0.4 1.1 ...
##  $ income  : num  11.74 9.32 9.95 10.66 9.73 ...
##  $ side    : Factor w/ 2 levels "n","s": 1 1 1 1 1 1 1 1 1 1 ...
```

```
model = lm(involact~race+fire+theft+age+log(income), chredlin)
summary(model)
```

```
##
## Call:
## lm(formula = involact ~ race + fire + theft + age + log(income),
##     data = chredlin)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85393 -0.16922 -0.03088  0.17890  0.81228
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.185540   1.100255  -1.078 0.287550
## race         0.009502   0.002490   3.817 0.000449 ***
## fire         0.039856   0.008766   4.547 4.76e-05 ***
## theft       -0.010295   0.002818  -3.653 0.000728 ***
## age          0.008336   0.002744   3.038 0.004134 **
## log(income)  0.345762   0.400123   0.864 0.392540
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3345 on 41 degrees of freedom
## Multiple R-squared:  0.7517, Adjusted R-squared:  0.7214
## F-statistic: 24.83 on 5 and 41 DF,  p-value: 2.009e-11
```

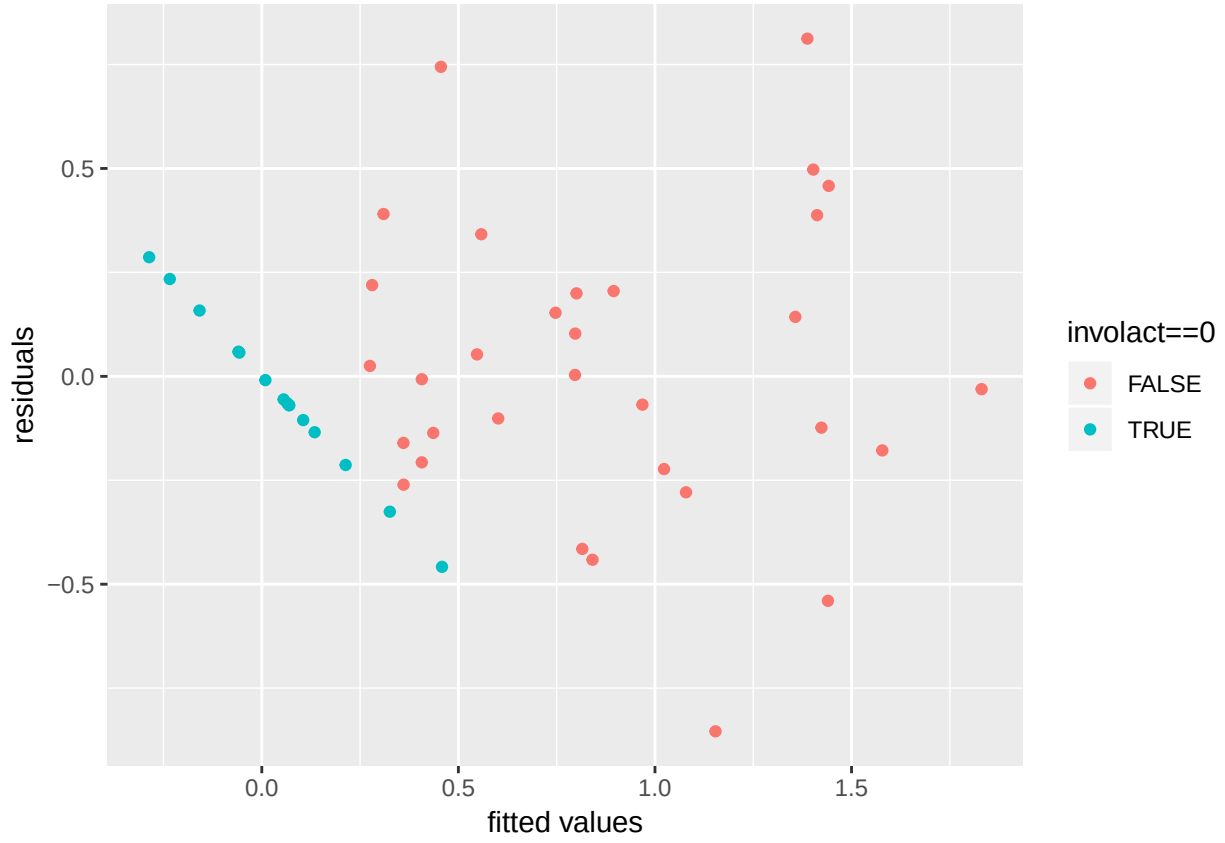
Answer:

The variable "log(income)" is not a significant predictor in this model, while the other four variables are all significant predictors. And the variables "race", "fire", "age" have positive relationships with the response "involact", while the variable "theft" has a negative relationship with the response "involact". Furthermore, the magnitude of all the coefficients of predictors are relatively small.

(c)

```
# plot residuals ~ fitted values
ggplot(chredlin, aes(x=model$fitted.values, y=model$residuals,
```

```
                    color=as.factor(involact==0))) +
  geom_point() + labs(x="fitted values", y="residuals", color="involact==0")
```



Answer:

The zero response values presents a linear line in the residuals against fitted values plot. This is because when the response "involact" = 0 (i.e. $y_i = 0$), the residuals $\hat{\epsilon_i} = y_i - \hat{y_i} = -\hat{y_i}$, so the plot for these points has a slope of -1.

Without theses points, the other points in the plot shows that the residuals are uncorrelated, normally distributed, and have nearly constant variance, so the Gaussian linear model assumptions are correct. However, including these points, the plot has a linear trend around the region of zero fitted values, indicating a nonconstant variance of residuals for having an issue of correlated residuals. This may mislead us to think that the Gaussian linear model assumptions are not correct and the model structure may have some problems.

(d)

The generalized linear model (GLM) for binary response here is:
likelihood: $P(involact_i = y_i|p_i) = p_i^{y_i}(1-p_i)^{1-y_i},\ y_i = 0, 1$
linear predictor: $\eta_i = \beta_0 + \beta_1 race_i + \beta_2 fire_i + \beta_3 theft_i + \beta_4 age_i + \beta_5 log(income_i) + \epsilon_i$
link function (logit): $\eta_i = log\frac{p_i}{1-p_i}$

```
# create a binary response variable for involact
chredlin$involact_b = as.numeric(chredlin$involact>0)

# fit a logistic regression model
model.glm = glm(involact_b~race+fire+theft+age+log(income), chredlin, family=binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model.glm)
```

```
##
## Call:
## glm(formula = involact_b ~ race + fire + theft + age + log(income),
##     family = binomial, data = chredlin)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -3.574e-04  -2.000e-08   2.000e-08   2.000e-08   4.421e-04
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.226e+04  1.059e+06   0.012    0.991
## race         3.042e+01  6.207e+03   0.005    0.996
## fire        -8.074e+01  1.931e+04  -0.004    0.997
## theft        1.656e+01  5.257e+03   0.003    0.997
## age          2.111e+00  4.943e+03   0.000    1.000
## log(income) -5.151e+03  3.860e+05  -0.013    0.989
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5.8865e+01  on 46  degrees of freedom
## Residual deviance: 4.9854e-07  on 41  degrees of freedom
## AIC: 12
##
## Number of Fisher Scoring iterations: 25
```

Answer:

In the summary results, none of the predictors is significant in this model, and the residual deviance is nearly zero. This is because of two problems: (1) "algorithm did not converge": the glm() uses an iteratively re-weighted least squares (IRLS) algorithm, and here the algorithm did not converge after the maximum number of allowed Fisher iterations, which is 25 by default. (2) "fitted probabilities numerically 0 or 1 occurred": there indicates the problem of a perfect fit in this model, which means that the fitted probabilities are extremely close to zero or one. residual deviance is zero indicating a perfect fit and yet none of the predictors are significant due to the high standard errors

This is probably due to complete separation, i.e. one group being entirely composed of 0s or 1s. The number of predictors is more than needed and causes overfitting problems, thus the two groups of the response variable are completely linearly separable. Also, in the cases of sparse data, both the complete separation and Hauck-Donner effect may occcur.

(e)

Now the linear predictor of the smaller GLM model is:
linear predictor: $\eta_i = \beta_0 + \beta_1 race_i + \beta_2 age_i + \epsilon_i$

```
# fit a smaller GLM model
model.glm2 = glm(involact_b~race+age, chredlin, family=binomial)
summary(model.glm2)
```

```
##
## Call:
```

```
## glm(formula = involact_b ~ race + age, family = binomial, data = chredlin)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.50010  -0.01286   0.00014   0.04390   1.69864
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -13.09746    7.44557  -1.759   0.0786 .
## race          0.32539    0.16602   1.960   0.0500 *
## age           0.14675    0.08794   1.669   0.0952 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 58.8653  on 46  degrees of freedom
## Residual deviance:  9.2286  on 44  degrees of freedom
## AIC: 15.229
##
## Number of Fisher Scoring iterations: 10
```

Answer:

The z-value, the test statistic of z-statistics, is $\hat{\beta}/se(\hat{\beta})$, which is approximately normally distributed. When the p-value of the z-statistics is smaller than the significance level α, then we reject the null hypothesis that the coefficient is zero, thus we think the coefficient is significant.

Here in this model summary, we see that the z-statistic of "race" is 1.960 and its p_value is 0.0500, so "race" is significant predictor at 5% significance level. On the other hand, the z-statistic of "age" is 1.669 and its p_value is 0.0952, so "age" is not a significant predictor at 5% significance level.

```
# Difference-in-deviances test for both predictors
LRT = model.glm2$null.deviance - model.glm2$deviance
df = model.glm2$df.null - model.glm2$df.residual
p_val = 1 - pchisq(LRT, df); p_val
```

```
## [1] 1.665479e-11
```

```
# Difference-in-deviances test for single predictors
drop1(model.glm2, test="Chi")
```

```
## Single term deletions
##
## Model:
## involact_b ~ race + age
##        Df Deviance    AIC    LRT Pr(>Chi)
## <none>       9.229 15.229
## race    1   45.408 49.408 36.179  1.8e-09 ***
## age     1   18.269 22.269  9.041  0.00264 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer:

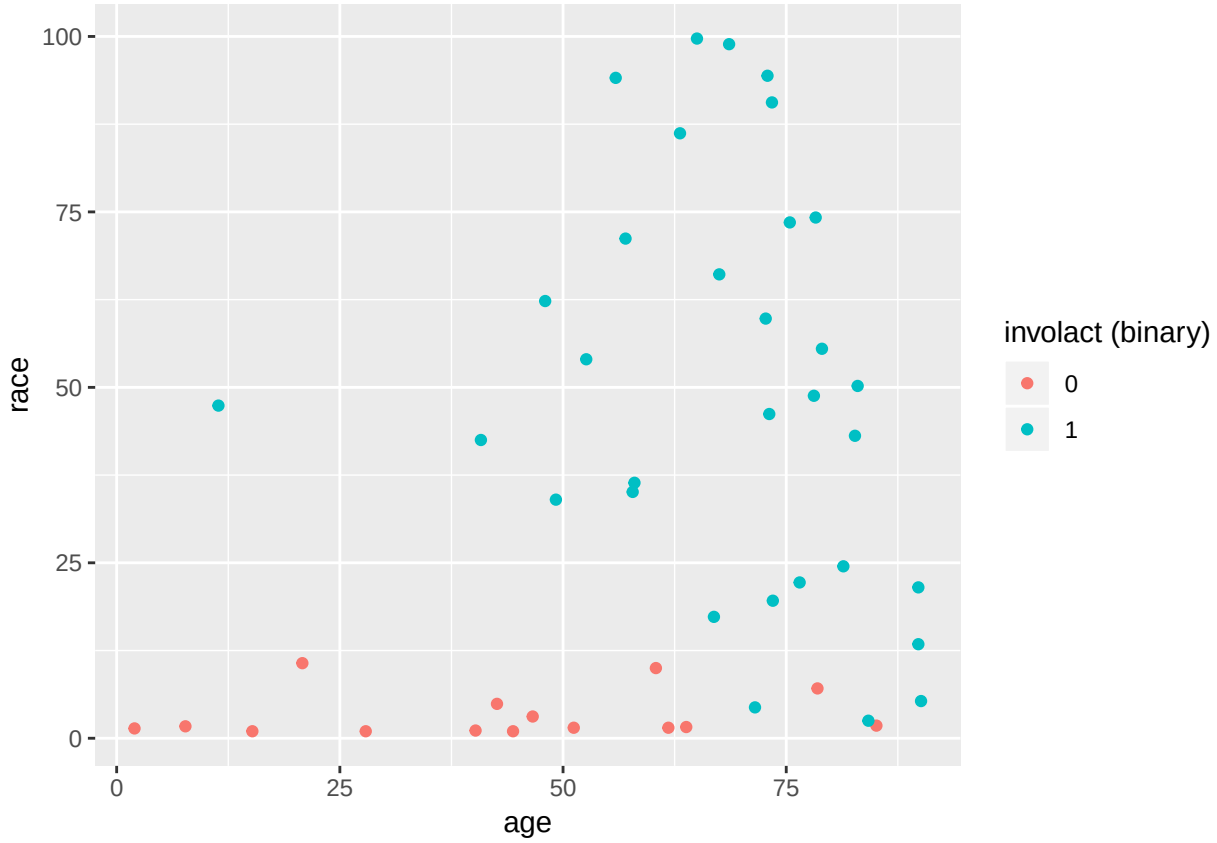
From the difference-in-deviance chi-square test, the results show that both "race" and "age" are significant predictors at 1% significance level, which is not the same with previous z-statistics results.

The difference-in-deviance chi-square test should be preferred for the significance of the predictors, because

this is a LRT (likelihood ratio test) that will not be affected by the sparse data effect. However, in some cases, especially with sparse data, the standard errors of z-statistics can be overestimated and so the z-value is too small, which makes the significance of the effect of a predictor could be missed. This is the so called Hauck-Donner effect. Therefore, the difference-in-deviance chi-square test is preferred.

(f)

```
# plot race ~ age
ggplot(chredlin, aes(x=age, y=race, color=as.factor(involact_b))) +
    geom_point() + labs(x="age", y="race", color="involact (binary)")
```



Answer:

The plot shows that when "race" is lower near zero value, the binary responses "involact" are nearly all equal to 0's as well, while "race" becomes larger than zero region, responses "involact" all become 1's.

This first indicates a significant positive relationship between "race" and "involact", which is consistent with the positive coefficient estimation of "race" and the result of the LRT significance test.

Second, this also makes the 0's and 1's of binary response "involact" are much easier to be separated. Here only with another one variable "age", in the two dimensional sample space, the binary response "involact" is not yet completely separable. However, this is already close to a complete separation. When adding more covariates like in the previous larger model with five predictors, in the higher dimensional sample space, this will lead to a complete separation case thus causing fitting problems.

On the other hand, for variable "age", we can also see that it has a positive relationship with "involact", but not as significant as "race". This is also consistent with the previous summary output and the LRT significance test.

(g)

```
# fit the binomial model with probit link
model.glm2_probit = glm(involact_b~race+age, chredlin, family=binomial(link=probit))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model.glm2_probit)
```

```
##
## Call:
## glm(formula = involact_b ~ race + age, family = binomial(link = probit),
##     data = chredlin)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.48896  -0.00040   0.00000   0.00789   1.65340
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.55984    4.16874  -1.813   0.0698 .
## race         0.18655    0.08913   2.093   0.0364 *
## age          0.08503    0.04939   1.722   0.0851 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 58.8653  on 46  degrees of freedom
## Residual deviance:  8.9786  on 44  degrees of freedom
## AIC: 14.979
##
## Number of Fisher Scoring iterations: 11
```

```
drop1(model.glm2_probit, test="Chi")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Single term deletions
##
## Model:
## involact_b ~ race + age
##        Df Deviance    AIC    LRT  Pr(>Chi)
## <none>       8.979 14.979
## race    1   45.550 49.550 36.572 1.472e-09 ***
## age     1   18.002 22.002  9.024  0.002665 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# compare the coefficients of two models
coef(model.glm2) / coef(model.glm2_probit)
```

```
## (Intercept)        race         age
##    1.732505    1.744306    1.725788
```

Answer:

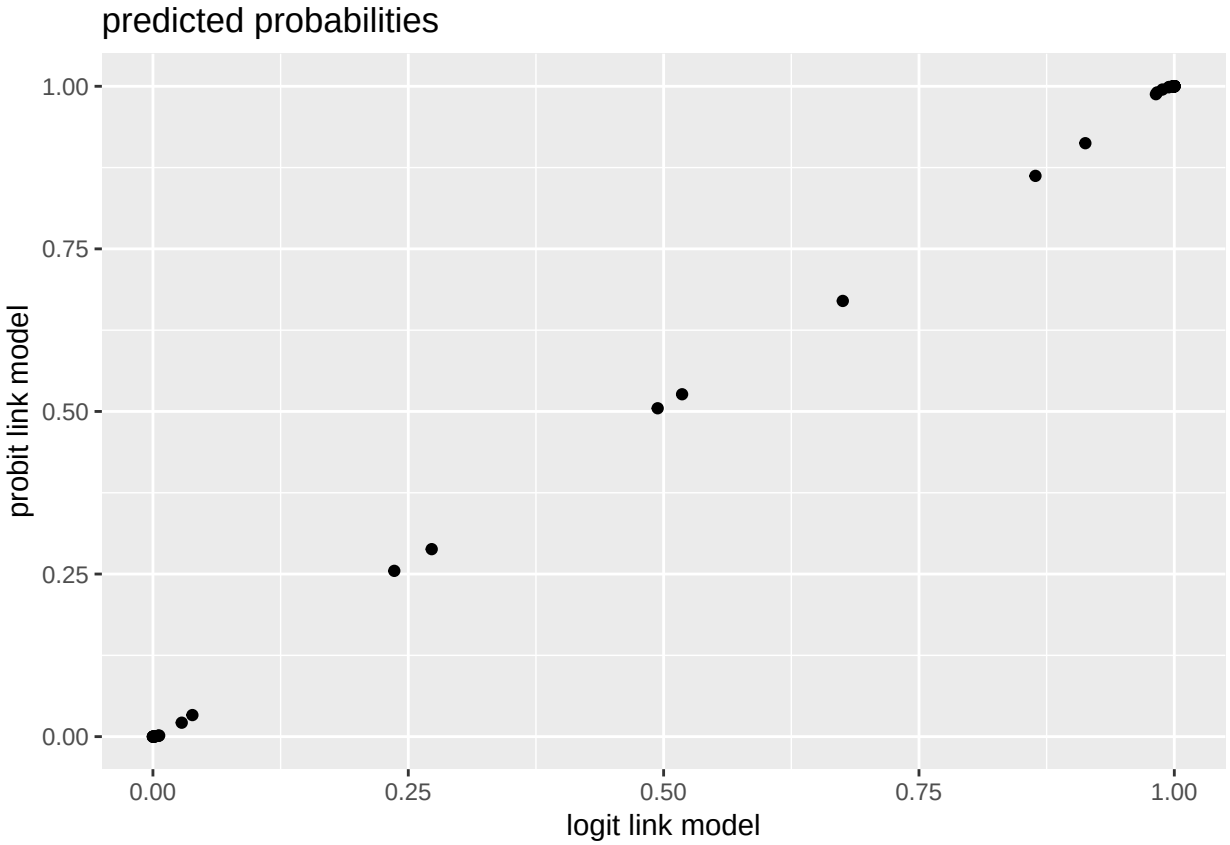
The signs of estimated coefficients are all the same in two models. Plus, their significance levels both in the

z-statistics and the difference-in-deviance LRT test statistics are the same for two models. What's more, the null deviances and the degree freedoms of the null models in two model summaries are the same.

However, the estimated coefficents in the logit model are all nearly 1.7 times larger than those in the probit model. And the residual deviances in two model summaries are different. On the other hand, the probit model has a warning sign that "fitted probabilities numerically 0 or 1 occurred" while the logit model is fine.

```
# plot predicted values on the probability scale
predprob_logit = predict(model.glm2, type="response")
predprob_probit = predict(model.glm2_probit, type="response")

ggplot(chredlin, aes(x=predprob_logit, y=predprob_probit)) +
    geom_point() +
    labs(x="logit link model", y="probit link model", title="predicted probabilities")
```




```
# find the relationships of them
m = lm(predprob_probit~predprob_logit)
summary(m)
```

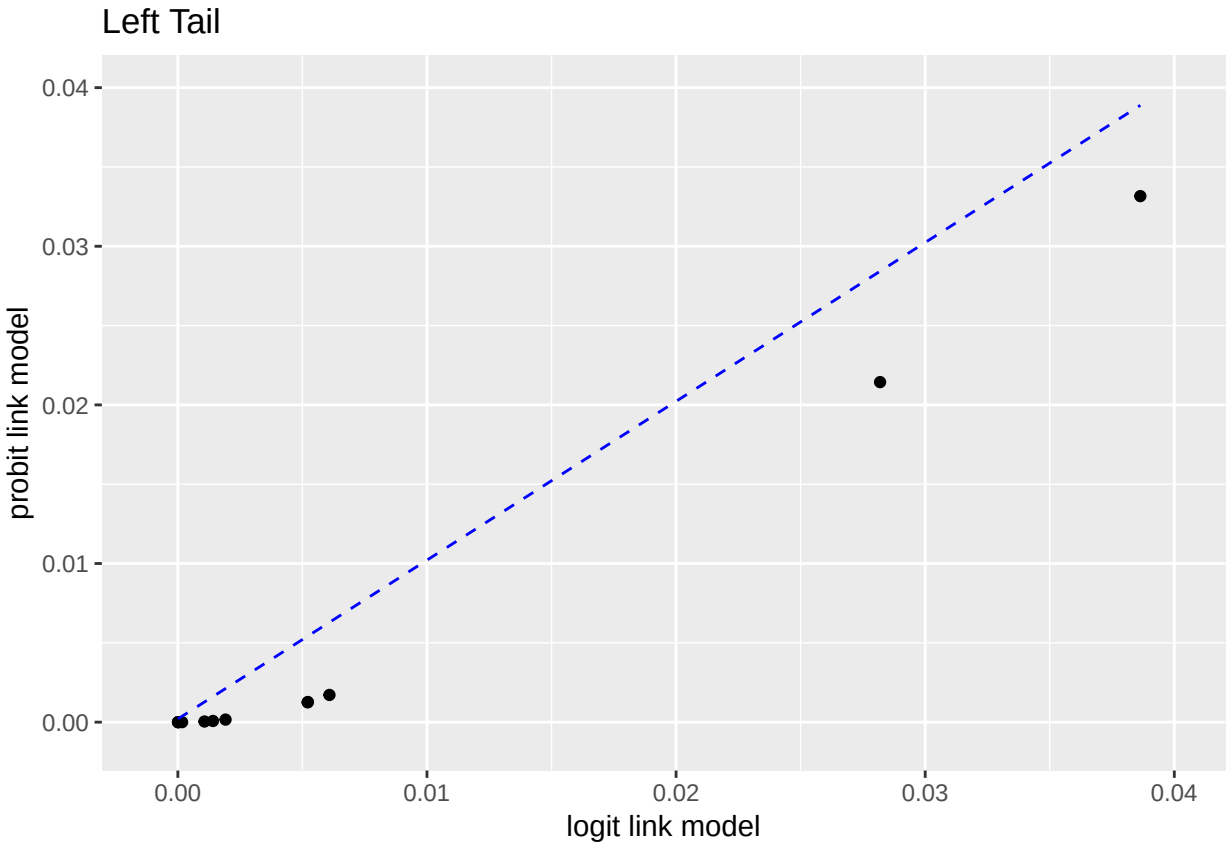
```
##
## Call:
## lm(formula = predprob_probit ~ predprob_logit)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0069886 -0.0014359 -0.0014352 -0.0002589  0.0181178
##
## Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.0002011  0.0013065   0.154    0.878
## predprob_logit 1.0012348  0.0016203 617.926   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.004799 on 45 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 3.818e+05 on 1 and 45 DF,  p-value: < 2.2e-16
```

```
# plot the two tails and the fitted line
ggplot(chredlin, aes(x=predprob_logit, y=predprob_probit)) +
    geom_point() + xlim(-0.001, 0.04) + ylim(-0.001, 0.04) +
    geom_line(aes(x=predprob_logit, y=m$fitted.values), linetype="dashed", color="blue") +
    labs(x="logit link model", y="probit link model", title="Left Tail")
```

```
## Warning: Removed 35 rows containing missing values (geom_point).
```

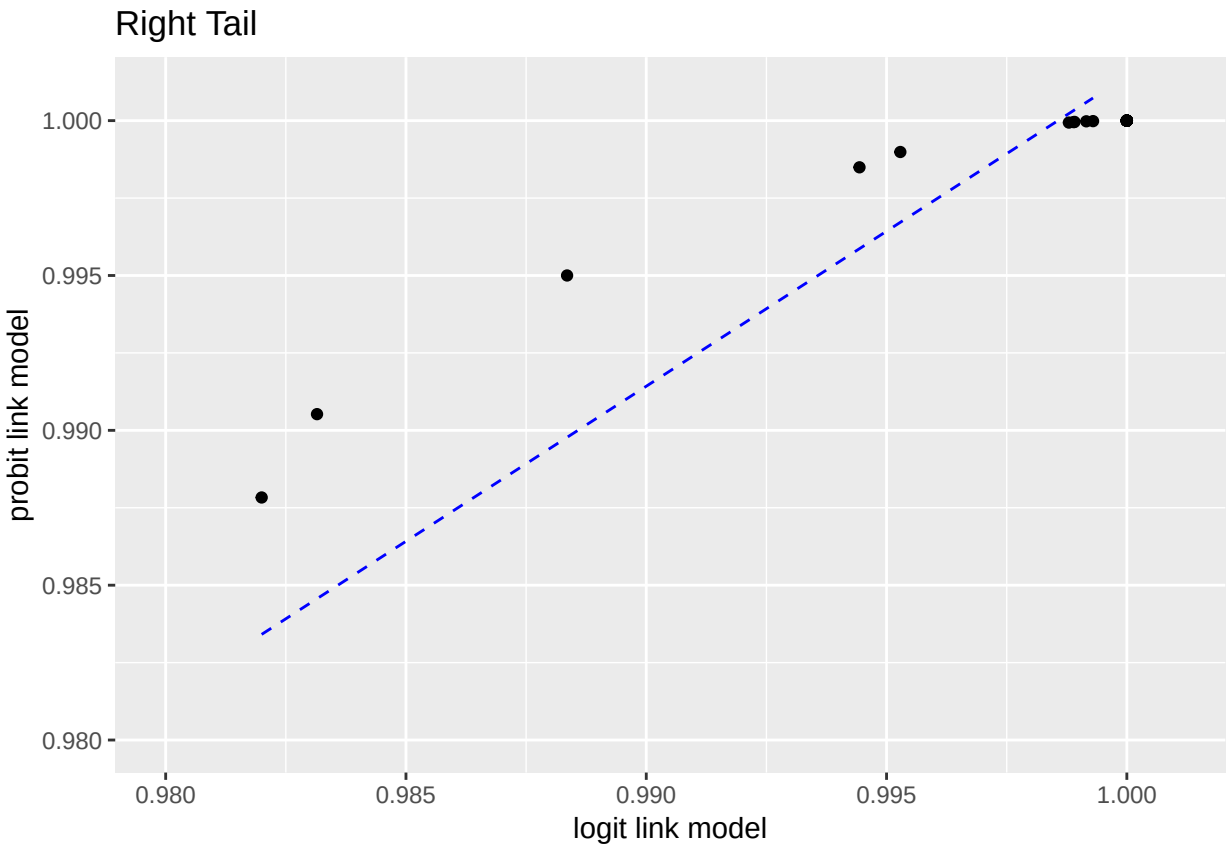
```
## Warning: Removed 35 rows containing missing values (geom_path).
```




```
ggplot(chredlin, aes(x=predprob_logit, y=predprob_probit)) +
    geom_point() + xlim(0.98, 1.001) + ylim(0.98, 1.001) +
    geom_line(aes(x=predprob_logit, y=m$fitted.values), linetype="dashed", color="blue") +
    labs(x="logit link model", y="probit link model", title="Right Tail")
```

```
## Warning: Removed 19 rows containing missing values (geom_point).
```

```
## Warning: Removed 37 rows containing missing values (geom_path).
```



Answer:

The plot shows that the predicted probabilites of two models are basically following a linear relationship with a significant slope of nearly 1. This means that the predicted probabilites of two models are basically the same, especially in the middle range of the predicted probabilites.

However, there seems to have some slightly differences in the two tails regions where the predictted probabilitis are near zero and one. In the left tail region, predicted probabilites of logit link model is slightly larger than that of probit link model. And in the right tail region, predicted probabilites of probit link model is slightly larger than that of logit link model.

(h)

Answer:

The logit link: $\eta = log\frac{p}{1-p} \quad \Rightarrow \quad p = \frac{e^{\eta}}{1+e^{\eta}}$

The probit link: $\eta = \Phi^{-1}(p) \quad \Rightarrow \quad p = \Phi(\eta) = P(Z \leq \eta)$

The logit model uses the cumulative distribution function of the logistic distribution, while the probit model uses the cumulatice distribution function of the standard normal distribution. Therefore, the problit link indicates a normally distributed latent variable in the model, so the probit link model for the binary response is most comparable to the Gaussian linear model, which also assumes that the response variable follows a normal distribution.

Both functions will take any number and rescale it to fall between 0 and 1, hence the linear predictor η can be transformed by the function to yield a predicted probability p. And bth methods will yield similar (though not identical) inferences, as shown in the previous questions.

However, the logit link model has slightly flatter tails, i.e. logit link model has a curve that approaches the axes slower than the that of probit link model.

On the other hand, logit link model is more popular in health sciences like epidemiology partly because its coefficients can be interpreted in terms of odds ratios thus giving better interpretations than the probit link model. And probit link model can be generalized to account for non-constant error variances in heteroskedastic probit models (e.g. in advanced econometric settings) and hence are used in some contexts by economists and political scientists.