Logistic Regression

Background

The owner of a company would like to be able to predict whether employees will stay with the company or leave. The data contains information about various characteristics of employees. See below for the description of these characteristics.

Data Description

The data consists of the following variables:

1. **Age.Group**: 1-9 (1 corresponds to teen, 2 corresponds to twenties, etc.) (numerical)
2. **Gender**: 1 if male, 0 if female (numerical)
3. **Tenure**: Number of years with the company (numerical)
4. **Num.Of.Products**: Number of products owned (numerical)
5. **Is.Active.Member**: 1 if active member, 0 if inactive member (numerical)
6. **Staying**: Fraction of employees that stayed with the company for a given set of predicting variables

Note: Please do not treat any variables as categorical.

Read the data

```
# import the data
data = read.csv("hw4_data.csv", header=TRUE, fileEncoding="UTF-8-BOM")
data$Staying = data$Stay/data$Employees
head(data)
```

	Age.Gro… <int>	Gen… <int>	Ten… <int>	Num.Of.Products <int>	Is.Active.Member <int>	S… <int>	Employe… <int>	Staying <dbl>
1	2	1	3	1	0	5	11	0.4545455
2	2	1	4	1	0	5	10	0.5000000
3	2	1	4	1	1	2	13	0.1538462
4	2	0	7	1	0	3	10	0.3000000
5	2	1	7	1	0	2	14	0.1428571
6	2	0	4	2	0	4	12	0.3333333

6 rows

```
dim(data)
```

```
## [1] 158   8
```

Question 1: Fitting a Model - 6 pts

Fit a logistic regression model using *Staying* as the response variable with *Num.Of.Products* as the predictor and logit as the link function. Call it **model1**.

(a) 2 pts - Display the summary of model1. What are the model parameters and estimates?

```
# Include weights since the response is proportion of success (and not binary 1, 0)
model1 = glm(Staying ~ Num.Of.Products, data = data, weights = Employees, family = binomial)
summary(model1)
```

```
##
## Call:
## glm(formula = Staying ~ Num.Of.Products, family = binomial, data = data,
##     weights = Employees)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2827  -1.4676  -0.1022   1.4490   4.7231
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)       2.1457     0.1318   16.27   <2e-16 ***
## Num.Of.Products  -1.7668     0.1031  -17.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 981.04  on 157  degrees of freedom
## Residual deviance: 632.04  on 156  degrees of freedom
## AIC: 1056.8
##
## Number of Fisher Scoring iterations: 4
```

Answer

Observe above model1's output. The model's parameters are β_0 or intercept and $\beta_{Num.Of.Products}$ and their estimations are 2.1457 and -1.7668, respectively.

(b) 2 pts - Write down the equation for the odds of staying.

Answer

$log(odds_{staying})$ = 2.1457-1.7668*$Num.Of.Products$

$odds_{staying}$ = $p_{staying}$ / (1 - $p_{staying}$) = $e^{2.1457-1.7668*Num.Of.Products}$

(c) 2 pts - Provide a meaningful interpretation for the coefficient for *Num.Of.Products* with respect to the log-odds of staying and the odds of staying.

Answer

$odds_{staying}$ = $p_{staying}$ / (1 - $p_{staying}$)

$P(Staying|Num.Of.Products)$ = $odds_{staying}$ / (1 + $odds_{staying}$)

As $Num.Of.Products$ increases by 1 unit, the log odds decreases by ***1.7668***, which is the same as the odds decreasing by a factor of $e^{-1.7668}$ (=***0.170878927***).

check:

$(log(odds)|Num.Of.Products = 1)$ = 2.1457 - 1.7668*1 = 0.3789

$(log(odds)|Num.Of.Products = 2)$ = 2.1457 - 1.7668*2 = -1.3879

0.3789 - ***1.7668*** = -1.3879

$(odds|Num.Of.Products = 1)$ = $e^{2.1457-1.7668*1}$ = 1.460676961

$(odds|Num.Of.Products = 2)$ = $e^{2.1457-1.7668*2}$ = 0.249598912

1.460676961*0.170878927 = 0.24959891

Question 2: Inference - 9 pts

(a) 3 pts - Using model1, find a 90% confidence interval for the coefficient for *Num.Of.Products*.

```
# Coefficient confidence interval

# Using confint function in R
CI_R = confint(model1, "Num.Of.Products", level=0.90)
```

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

```
# Calculating manually
z = 1.645
b = -1.7668
se = 0.1031

ci = z*se
ci_low = b - ci
ci_up = b + ci

print("confidence interval using confint in R: ")
```

```
## [1] "confidence interval using confint in R: "
```

```
print(CI_R)
```

```
##       5 %      95 %
## -1.938361 -1.598965
```

```
print("confidence interval calculated manually: ")
```

```
## [1] "confidence interval calculated manually: "
```

```
print(paste("lower bound calculated manually: ", ci_low))
```

```
## [1] "lower bound calculated manually:  -1.9363995"
```

```
print(paste("upper bound calculated manually: ", ci_up))
```

```
## [1] "upper bound calculated manually:  -1.5972005"
```

Answer

See above the 90% confidence interval for $Num.Of.Products$ using the confint function in R and calculated manually. Note that there is a slight difference between the two calculation. The confint function gives an interval of [-1.938361, -1.598965] and the manual calculation gives an interval of [-1.9363995, -1.5972005].

(b) 3 pts - Is model1 significant overall? How do you come to your conclusion?

```
# Test for overall regression
gstat = model1$null.deviance - deviance(model1)
1-pchisq(gstat, length(coef(model1))-1)
```

```
## [1] 0
```

Answer

The overall model is significant. See above the overall regression test where we check for the null hypothesis where $\beta_{Num.Of.Products}$ = 0. With a p-value of 0, we can reject the null hypothesis and conclude that $\beta_{Num.Of.Products}$ is not 0 and the overall regression is significant.

(c) 3 pts - Which coefficients are significantly nonzero at the 0.01 significance level? Which are significantly negative? Why?

```
confint(model1, level=0.99)
```

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

```
##                     0.5 %    99.5 %
## (Intercept)      1.809088  2.488638
## Num.Of.Products -2.037227 -1.505437
```

Answer

Both the $intercept$ and $Num.Of.Products$ are significantly nonzero at the 0.01 significance level since both of their p-values (2e-16 and 2e-16) are smaller than 0.01 and since the coefficients confidence intervals do not include zero. However, only the $Num.Of.Products$ is significantly negative based on the 99% confidence interval provided above. The upper and lower bound of the $intercept$ are positive and the interval does not include zero which means the coefficient is significantly positive. However, the upper and lower bound of $Num.Of.Products$ are negative and the interval does not include zero which means the coefficient is significantly negative.

Question 3: Goodness of fit - 9 pts

(a) 3.5 pts - Perform goodness of fit hypothesis tests using both deviance and Pearson residuals. What do you conclude? Explain the differences, if any, between these findings and what you found in Question 2b.

```
## Test for GOF: Using deviance residuals
c(deviance(model1), 1-pchisq(deviance(model1),model1$df.residual))
```

```
## [1] 632.04   0.00
```

```
## Test for GOF: Using Person residuals
pearres = residuals(model1,type="pearson")
pearson.tvalue = sum(pearres^2)
c(pearson.tvalue, 1-pchisq(pearson.tvalue,model1$df.residual))
```

```
## [1] 562.1763   0.0000
```

Answer

Here the null hypothesis is that the model fits the data. Here, we actually looking for a large p-value in order NOT to reject the null. As indicated in the code's output above, based on both deviance and Pearson residuals tests, the p-value is 0, hence we reject the null and conclude the model does not fit the data well.

In Question 2b we tested for the significance of the overall model or whether the coefficient had predictive power and here we tested for goodness of fit. These two tests provide different inferences about the model. The overall test provides inferences on the predictive power of the model whereas the goodness of fit test provides inferences on the goodness of fit of the model. Goodness of fit means that the model assumptions hold. Predictive power means that the predicting variables predict the data even if one or more of the assumptions do not hold. This means that we could have a model that predicts well but does not fit the data well or vice versa.

(b) 3.5 pts - Perform visual analytics for checking goodness of fit for this model and write your observations. Be sure to address the model assumptions. Only deviance residuals are required for this question.

```
# Residuals analysis
library(ggplot2)

# Linearity assumption
odds = data$Staying / (1 - data$Staying) # Calculating odds. "Staying" is the probability and we need to plot the log(odds) vs. Predictor to asses Linearity assumption.
Num.Of.Products = data$Num.Of.Products
odds_df = data.frame(odds, Num.Of.Products)

ggplot(odds_df, aes(x=Num.Of.Products, y=log(odds))) + ggtitle("Num.Of.Products vs. Log(odds)")
 + geom_point() +
  scale_colour_hue(l=50) + # Use a slightly darker palette than normal
  geom_smooth(method="loess",   # Add regression line
              se=FALSE,    # Don't add shaded confidence region
              fullrange=TRUE) # Extend regression lines
```

```
## `geom_smooth()` using formula 'y ~ x'
```

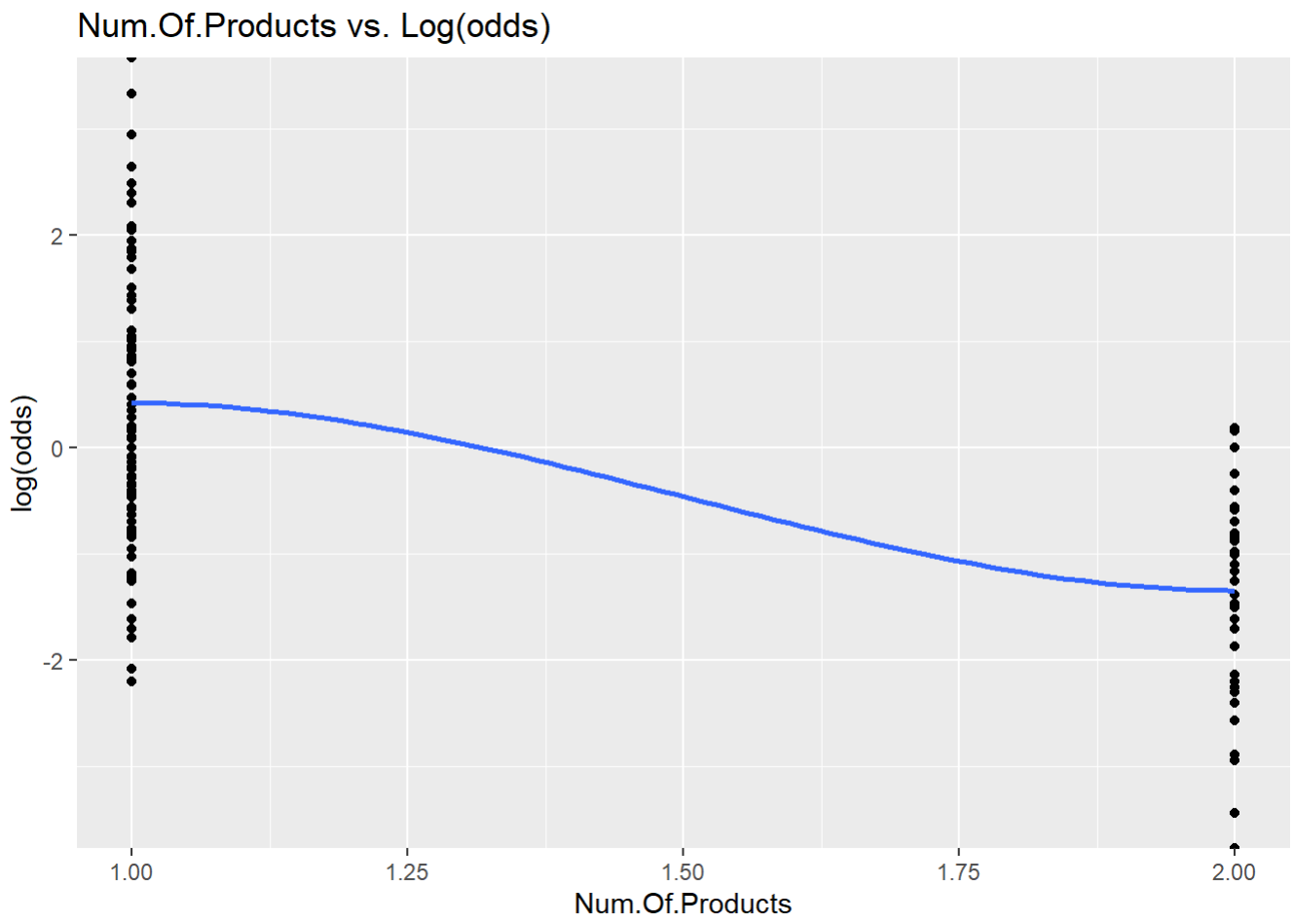
```
## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.995
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```




```
# plot(Num.Of.Products,log(odds), ylab="Logit of Staying",
#      main="Scatterplot of Num.Of.Products vs. Log(odds)")

# Independence assumption
res = resid(model1,type="deviance")

res_df = data.frame(res, Num.Of.Products)
ggplot(res_df, aes(x=Num.Of.Products, y=res)) + ggtitle("Num.Of.Products vs. Residuals (devianc
e)") + geom_point() + annotate("text", x= 1.5, y=4.5, label= mean(res), col="red") +
  annotate("text", x= 1.5, y=5, label= "Residulas Mean", col="red") +
  scale_colour_hue(l=50) + # Use a slightly darker palette than normal
  geom_smooth(method="loess",   # Add regression line
              se=FALSE,    # Don't add shaded confidence region
              fullrange=TRUE) # Extend regression lines
```

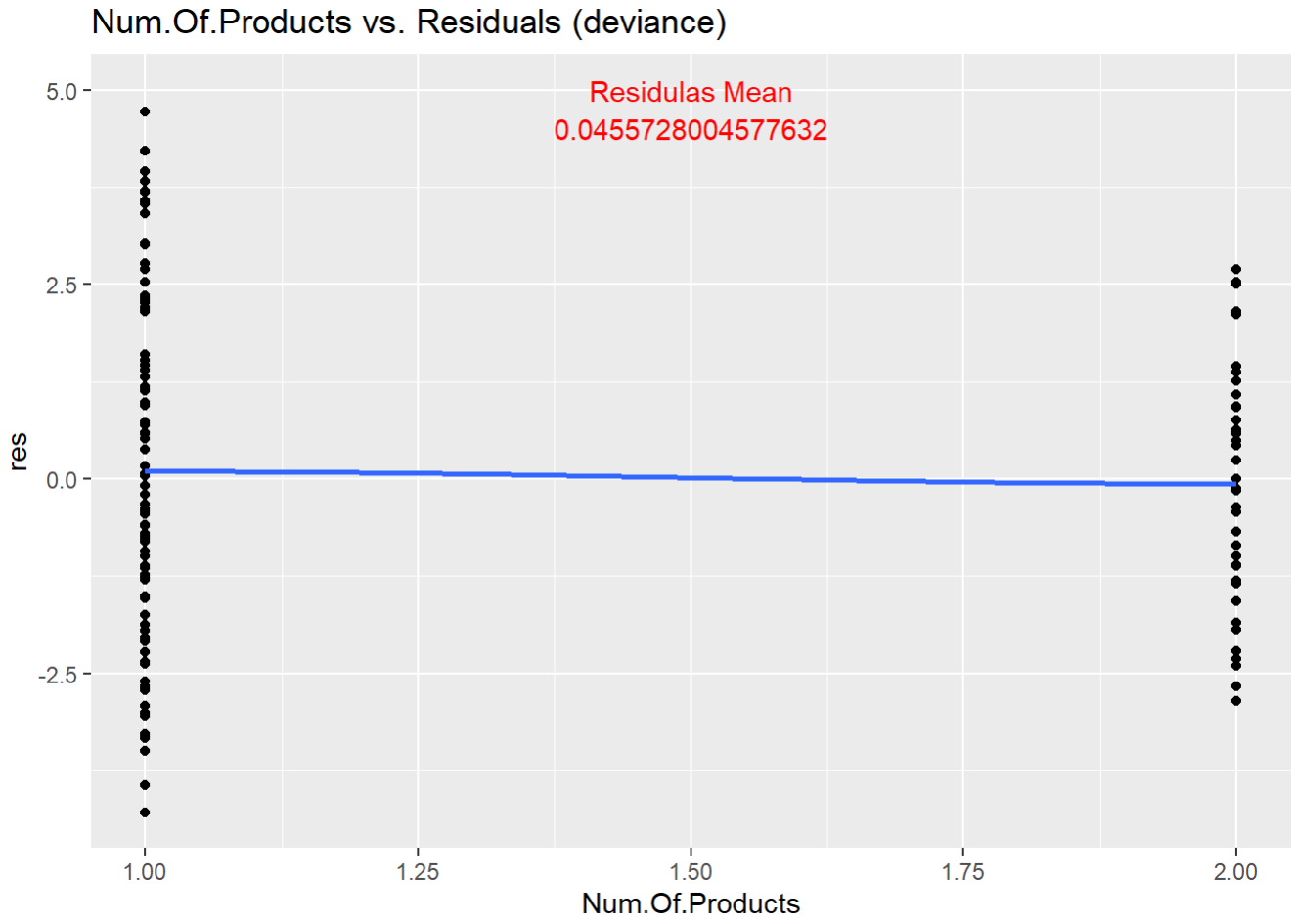
```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.995
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

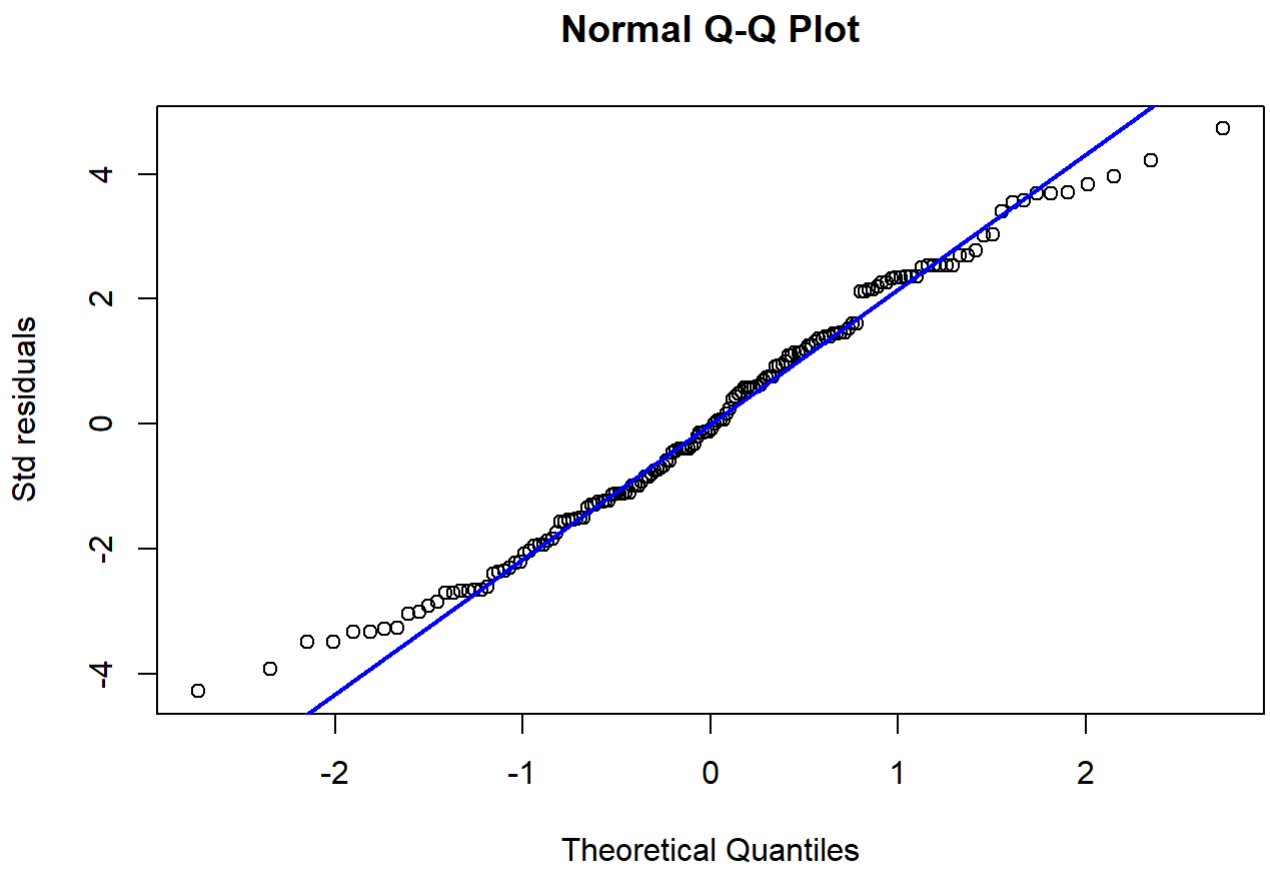
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```

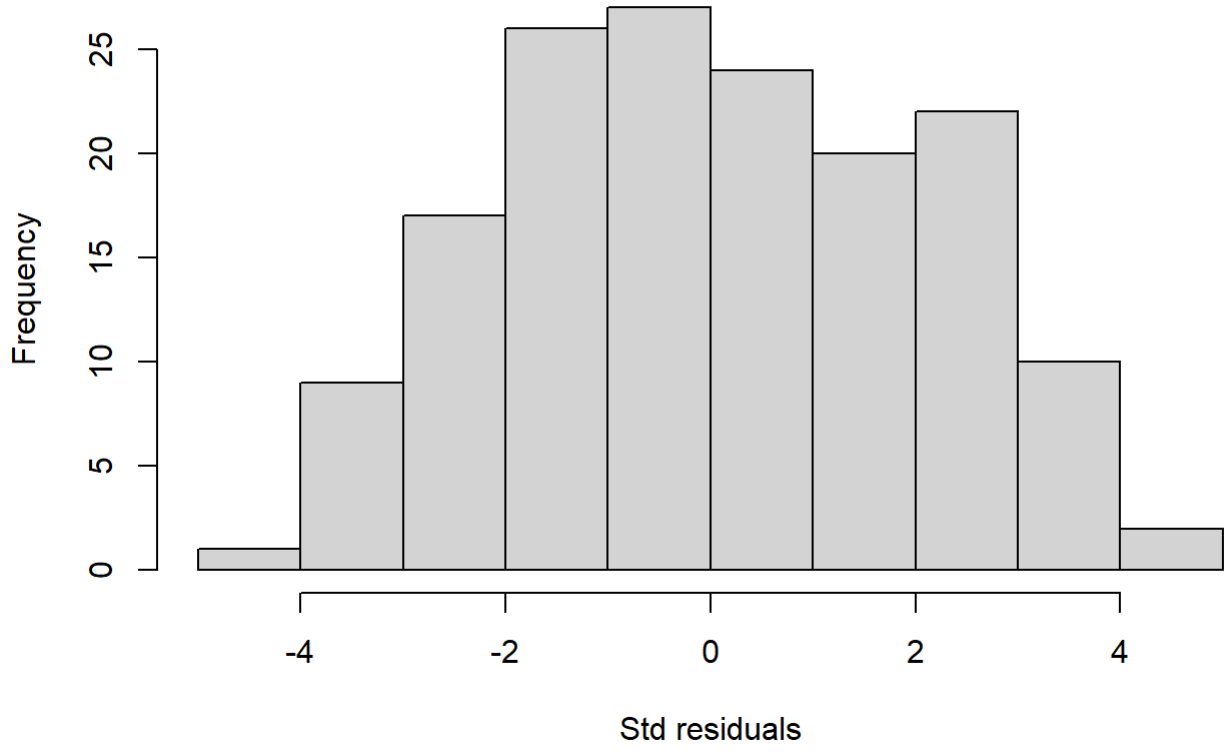



```
# plot(Num.Of.Products, res, main="Num.Of.Product vs. Residuals (deviance)")
# abline(h=mean(res), lwd=2, col="blue")
# legend("topright", legend=c("mean of residuals"),
#        col=c("blue"), lty=1:2, cex=0.9)

# Residuals Normality
qqnorm(res, ylab="Std residuals")
qqline(res,col="blue",lwd=2)
```




```
hist(res,xlab="Std residuals", main="")
```




```
# Hypothesis testing for Normality of residuals
# Null hypothesis is the data is normal distributed. Want LARGE p-value in order NOT to reject.
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.98455, p-value = 0.07577
```

Answer

The Logistic Regression's assumptions are:

Linearity Assumption - $g(p(X_1, \ldots, X_p)) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$

Independence Assumption - $Y_1, \ldots, Y_n$ are independent random variables.

Logit Link Function - $g(p) = ln(\frac{p}{1-p}) = ln(odds)$

Looking at the first plot (Num.Of.Products vs. Log(odds)), we can see the linearity assumption holds since there is somewhat linear relationship between the predictor $Num.Of.Products$ and the $Log(odds)$. Testing for the independence assumption we plot the residuals against the predictor and we can see the residuals are uncorrelated (though it is hard to evaluate given Num.Of.Products has only two values). Also, assuming the model is a good fit, the residuals should be normal distributed with a mean 0. In the Num.Of.Products vs. Residuals

(deviance) plot we can see the residuals mean is 0.04557 which is ~ 0. The Q-Q Normal and Histogram plots are shown as heavy-tailed indicating the normality of the residuals assumption does not hold. The Shapiro-Wilk test for normality however indicates the residuals are normally distributed. The null hypothesis of the Shapiro-Wilk test is that the data is normal distributed and with a p-value of 0.07577, we cannot reject the null at α = 0.05 and conclude the residuals are normally distributed. Note however that the Shapiro-Wilk test's p-value is not so large and barely rejects the null.

(c) 2 pts - Calculate the dispersion parameter for this model. Is this an overdispersed model?

```
D = deviance(model1)
DF = model1$df.residual
D
```

```
## [1] 632.04
```

```
DF
```

```
## [1] 156
```

```
D/ DF
```

```
## [1] 4.051539
```

Answer

The dispersion parameter ϕ is calculated as the sum of the squared deviance divided by the degrees of freedom $n - p - 1$, which in model1 is ϕ = 632.04 / 156 = 4.051539. Since ϕ > 2, we can conclude the model is overdispersed.

Question 4: Fitting the full model- 20 pts

Fit a logistic regression model using *Staying* as the response variable with *Age.Group*, *Gender*, *Tenure*, *Num.Of.Products*, and *Is.Active.Member* as the predictors and logit as the link function. Call it **model2**.

```
model2 = glm(Staying ~ Age.Group+Gender+Tenure+Num.Of.Products+Is.Active.Member, weights=Employe
es, data = data, family = binomial)
```

(a) 2.5 pts - Write down the equation for the probability of staying.

```
summary(model2)
```

```
##
## Call:
## glm(formula = Staying ~ Age.Group + Gender + Tenure + Num.Of.Products +
##     Is.Active.Member, family = binomial, data = data, weights = Employees)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2638  -0.7662   0.0018   0.6836   2.8912
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.903330   0.330549  -5.758 8.51e-09 ***
## Age.Group         1.229014   0.075158  16.352  < 2e-16 ***
## Gender           -0.551438   0.093139  -5.921 3.21e-09 ***
## Tenure           -0.003574   0.016470  -0.217    0.828
## Num.Of.Products  -1.428767   0.111181 -12.851  < 2e-16 ***
## Is.Active.Member -0.871460   0.095034  -9.170  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 981.04  on 157  degrees of freedom
## Residual deviance: 171.94  on 152  degrees of freedom
## AIC: 604.66
##
## Number of Fisher Scoring iterations: 4
```

Answer

$odds_{staying}$ = $p_{staying}$ / (1 - $p_{staying}$) = $e^{-1.903330+1.229014*Age.Group-0.551438*Gender-0.003574*Tenure-1.428767*Num.Of.Products-0.871460*Is.Active.Member}$

P(Staying | the predicting variables above) = $odds$ / (1 + $odds$) = $e^{-1.903330+1.229014*Age.Group-0.551438*Gender-0.003574*Tenure-1.428767*Num.Of.Products-0.871460*Is.Active.Member}$ / (1 + $e^{-1.903330+1.229014*Age.Group-0.551438*Gender-0.003574*Tenure-1.428767*Num.Of.Products-0.871460*Is.Active.Member}$)

(b) 2.5 pts - Provide a meaningful interpretation for the coefficients of *Age.Group* and *Is.Active.Member* with respect to the odds of staying.

Answer

Holding all other predicting variables fixed,

as $Age.Group$ increases by 1 unit, the $odds$ increases by a factor of $e^{1.229014}$ = ***3.417857866*** and

as $Is.Active.Member$ increases by 1 unit, the $odds$ decreases by a factor of $e^{-0.871460}$ = ***0.418340326***

check (Holding all other predicting variables fixed):

Age.Group

$(odds|Age.Group = 1)$ = $e^{-1.903330+1.229014*1}$ = 0.509504802

$(odds|Age.Group = 2)$ = $e^{-1.903330+1.229014*2}$ = 1.741414998

0.509504802* ***3.417857866*** = 1.74141499

Is.Active.Member

$(odds|Is.Active.Member = 0)$ = $e^{-1.903330-0.871460*0}$ = 0.149071384

$(odds|Is.Active.Member = 1)$ = $e^{-1.903330-0.871460*1}$ = 0.062362571

0.149071384* ***0.418340326*** = 0.062362571

(c) 2.5 pts - Is *Is.Active.Member* significant given the other variables in model2?

Answer

Given the other predictors in the model, $Is.Active.Member$ is significant (at the level α = 0.01) because its p-value of 2e-16 is approximately 0 and smaller than 0.01.

(d) 10 pts - Has your goodness of fit been affected? Repeat the tests, plots, and dispersion parameter calculation you performed in Question 3 with model2.

Goodness of fit - Hypothesis testing

```
## Test for GOF: Using deviance residuals
print("Test for GOF: Using deviance residuals. chi_statistic and p-value:")
```

```
## [1] "Test for GOF: Using deviance residuals. chi_statistic and p-value:"
```

```
c(deviance(model2), 1-pchisq(deviance(model2),model2$df.residual))
```

```
## [1] 171.9381966   0.1282109
```

```
## Test for GOF: Using Person residuals
pearres = residuals(model2,type="pearson")
pearson.tvalue = sum(pearres^2)
print("Test for GOF: Using Person residuals. chi_statistic and p-value:")
```

```
## [1] "Test for GOF: Using Person residuals. chi_statistic and p-value:"
```

```
c(pearson.tvalue, 1-pchisq(pearson.tvalue,model2$df.residual))
```

```
## [1] 166.390888   0.200838
```

Goodness of fit - Visual analysis

```
# Residuals analysis plots

# Linearity assumption
odds = data$Staying / (1 - data$Staying) # Calculating odds. "Staying" is the probability and we
need to plot the log(odds) vs. Predictor to asses Linearity assumption.
Age.Group = data$Age.Group
Gender = data$Gender
Tenure = data$Tenure
Num.Of.Products = data$Num.Of.Products
Is.Active.Member = data$Is.Active.Member
odds_df2 = data.frame(Age.Group, Gender, Tenure, Num.Of.Products, Is.Active.Member, odds)

predictors = c("Age.Group", "Gender", "Tenure", "Num.Of.Products", "Is.Active.Member")
n = length(predictors)

for (i in c(1:n)){
  print(ggplot(odds_df2, aes(x=odds_df2[,i], y=log(odds))) + ggtitle(paste(predictors[i], "vs. L
og(odds)")) + xlab(predictors[i]) + geom_point() +
    scale_colour_hue(l=50) + # Use a slightly darker palette than normal
    geom_smooth(method="loess",   # Add regression line
                se=FALSE,    # Don't add shaded confidence region
                fullrange=TRUE)) # Extend regression lines
}
```

```
## `geom_smooth()` using formula 'y ~ x'
```

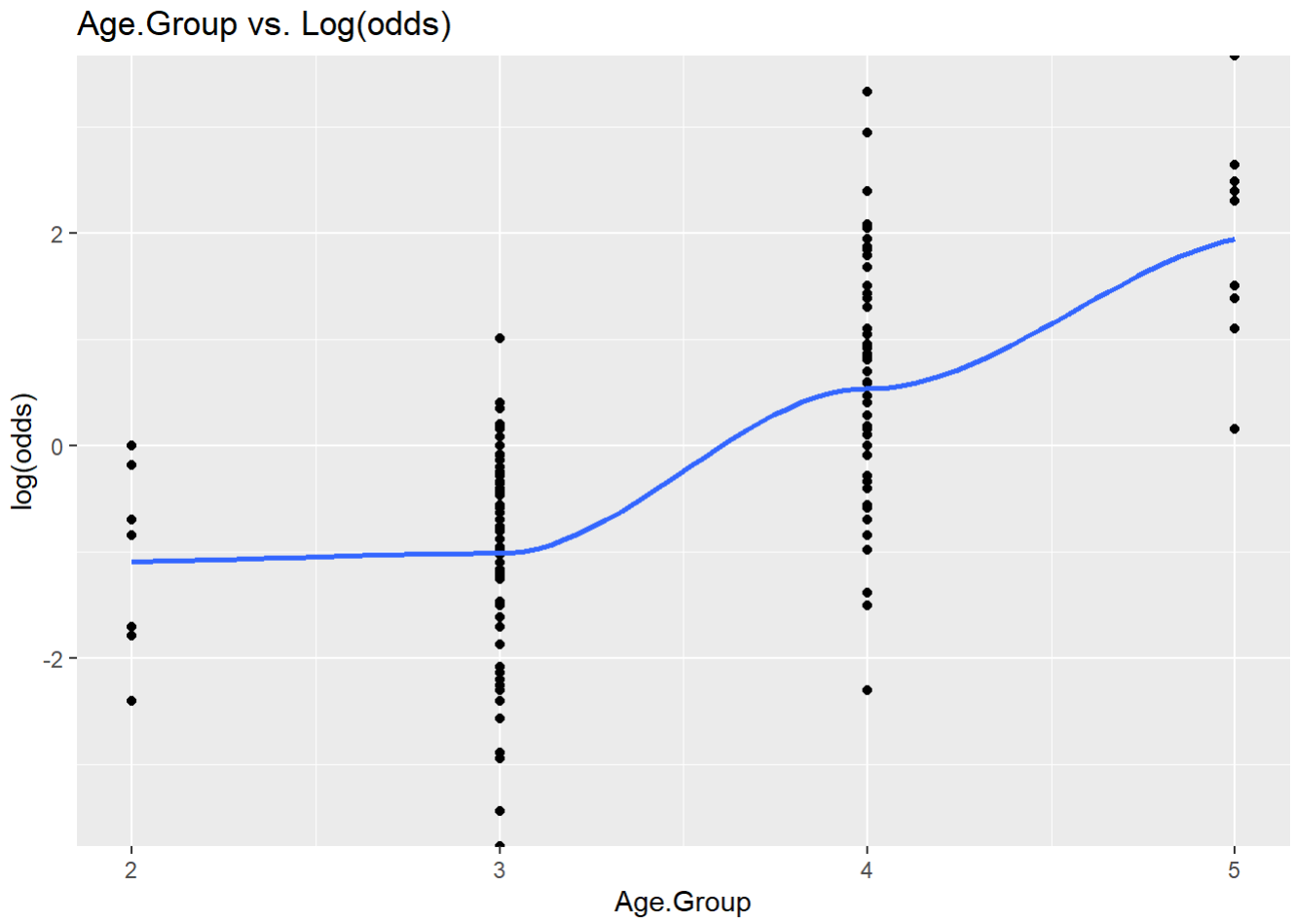
```
## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1.985
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 2.015
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 5.0803e-016
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 4.0602
```




```
## `geom_smooth()` using formula 'y ~ x'
```

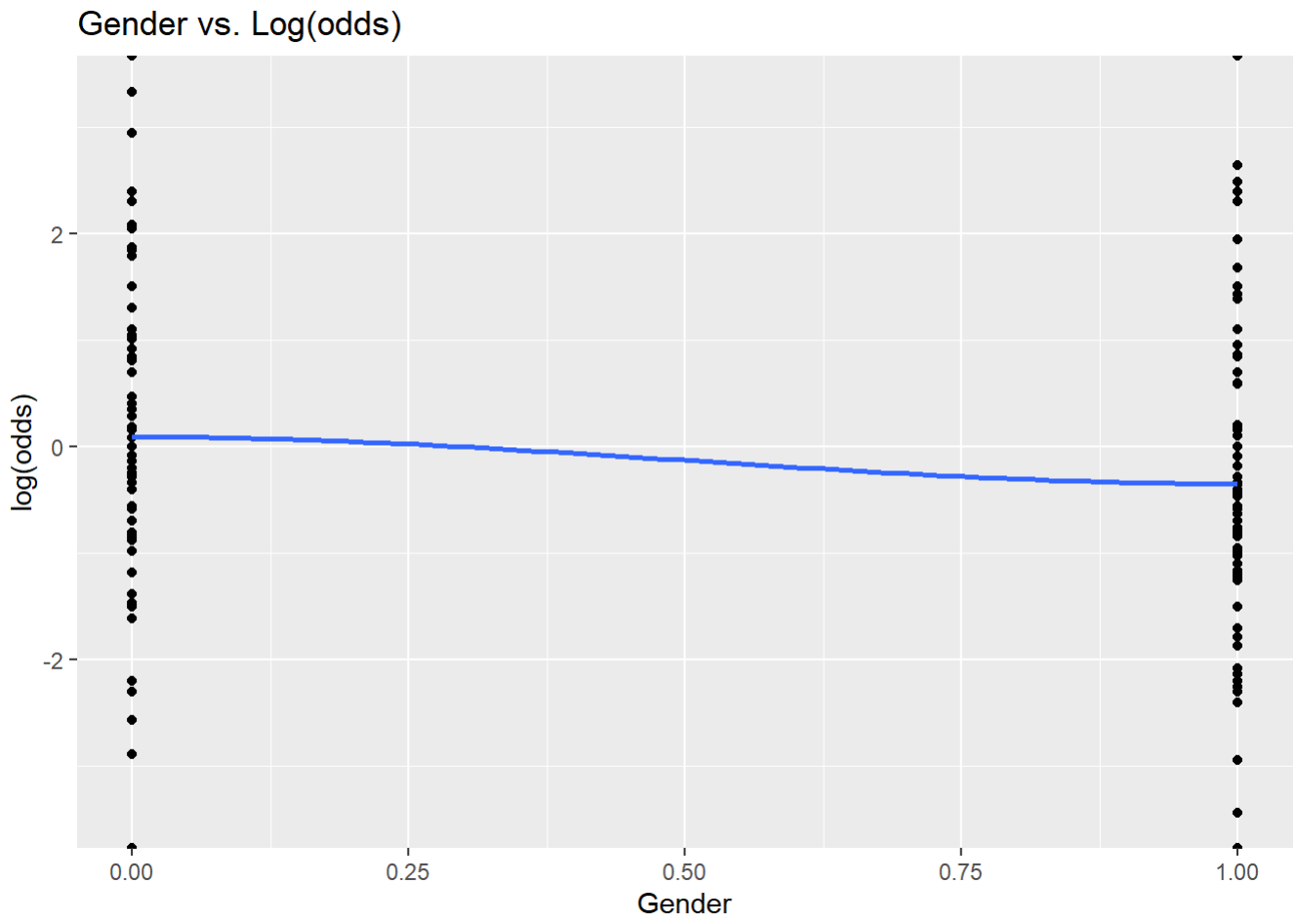
```
## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

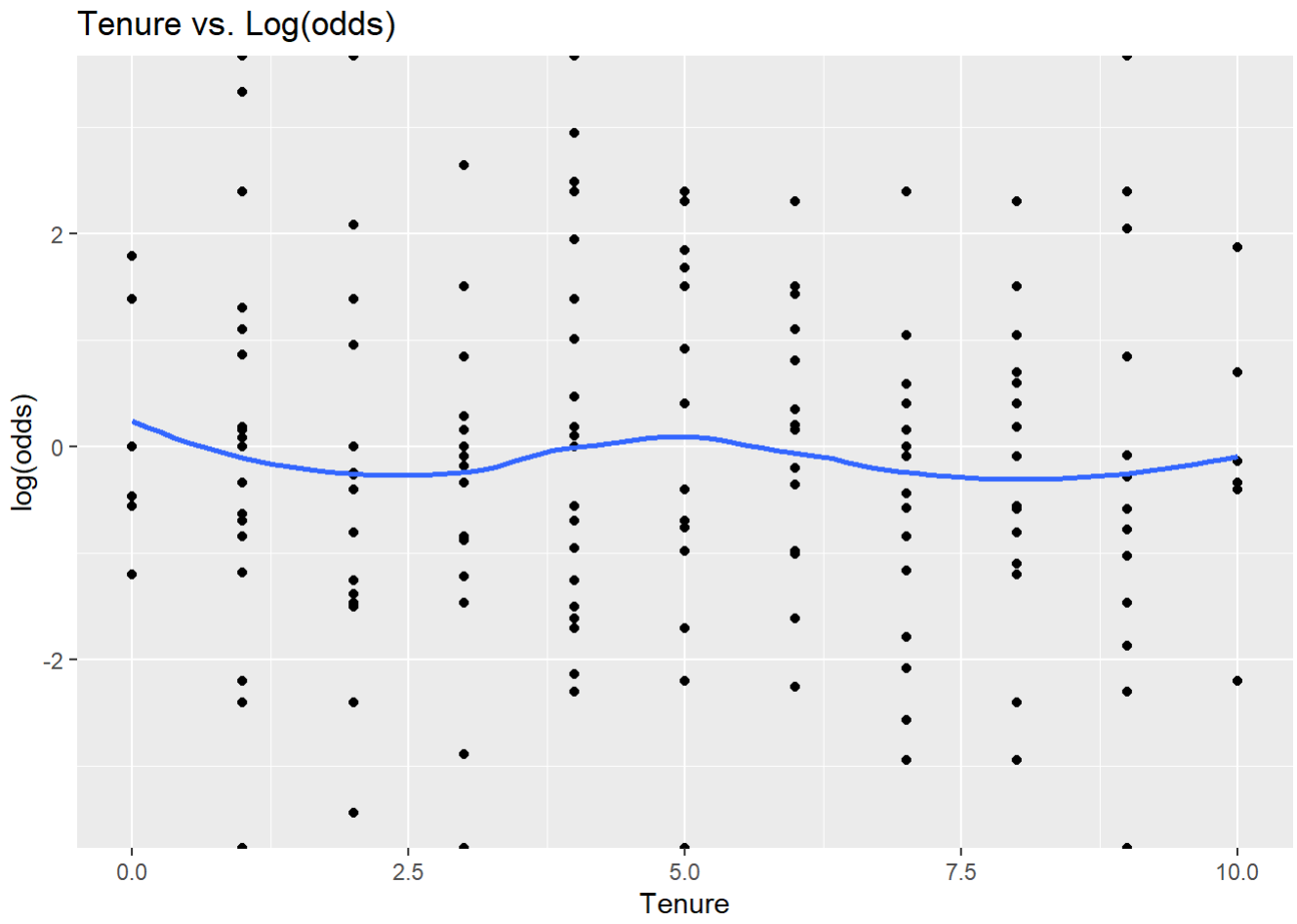
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```




```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```




```
## `geom_smooth()` using formula 'y ~ x'
```

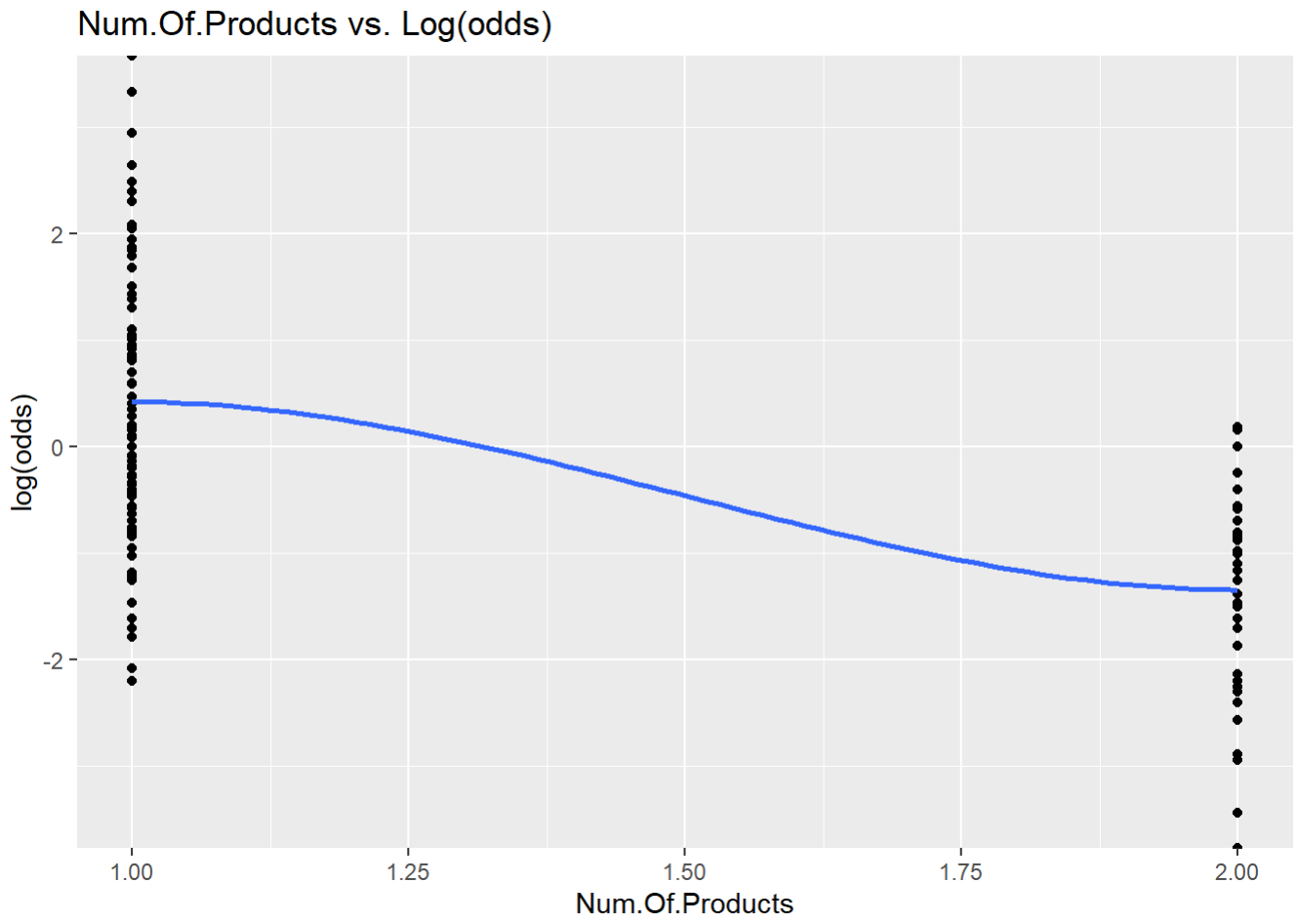
```
## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.995
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```




```
## `geom_smooth()` using formula 'y ~ x'
```

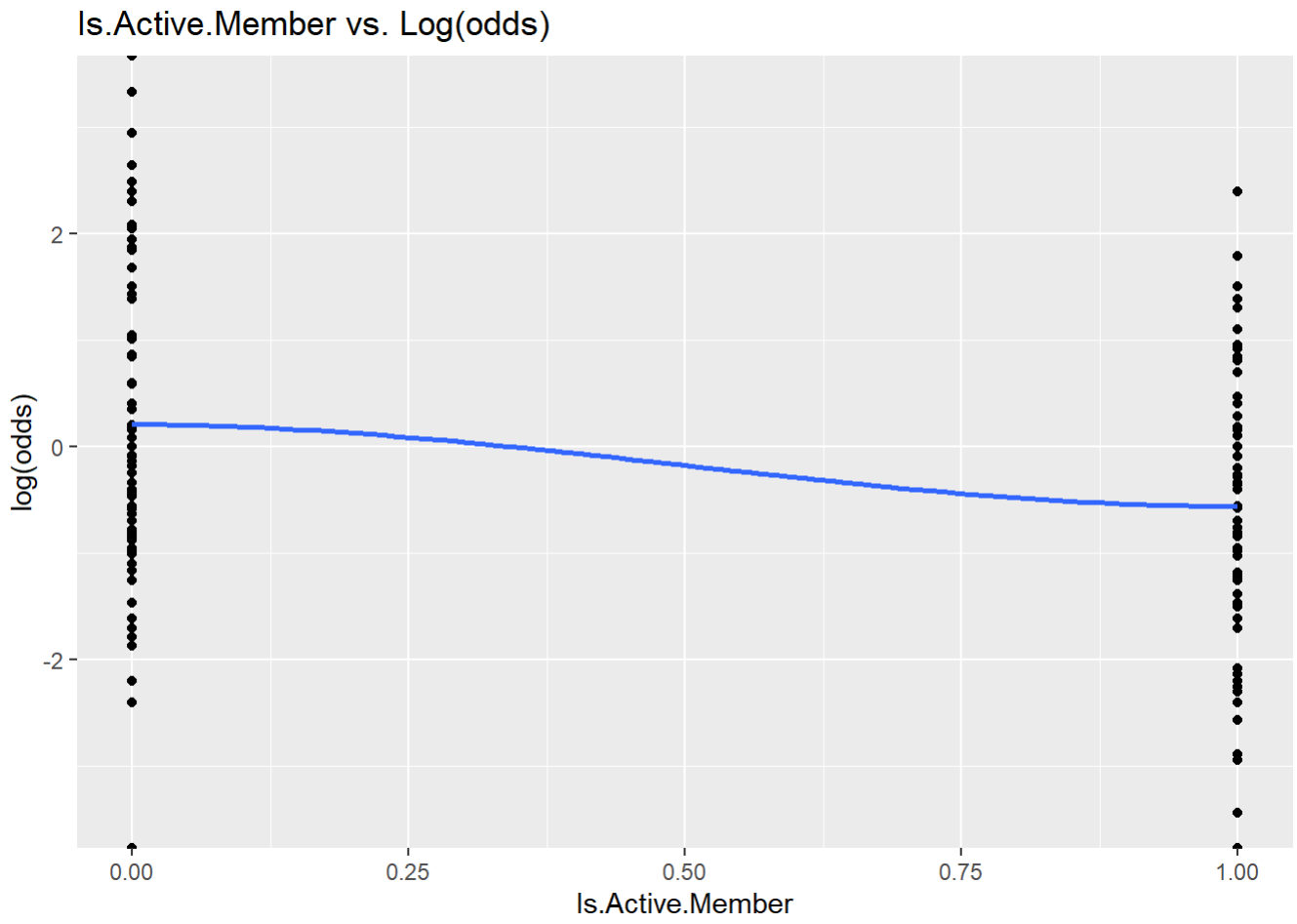
```
## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 7.1561e-031
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```




```
# Independence assumption
res2 = resid(model2,type="deviance")

res_df2 = data.frame(Age.Group, Gender, Tenure, Num.Of.Products, Is.Active.Member, res2)
for (i in c(1:n)){
print(ggplot(res_df2, aes(x=res_df2[,i], y=res2)) + ggtitle(paste(predictors[i], "vs. Residuals
 (deviance)")) + xlab(predictors[i]) + geom_point() +
  annotate("text", x= mean(res_df2[,i]), y=max(res2)-0.5, label= mean(res2), col="red") +
  annotate("text", x= mean(res_df2[,i]), y=max(res2), label= "Residulas Mean", col="red") +
  scale_colour_hue(l=50) + # Use a slightly darker palette than normal
  geom_smooth(method="loess",   # Add regression line
              se=FALSE,    # Don't add shaded confidence region
              fullrange=TRUE)) # Extend regression lines
}
```

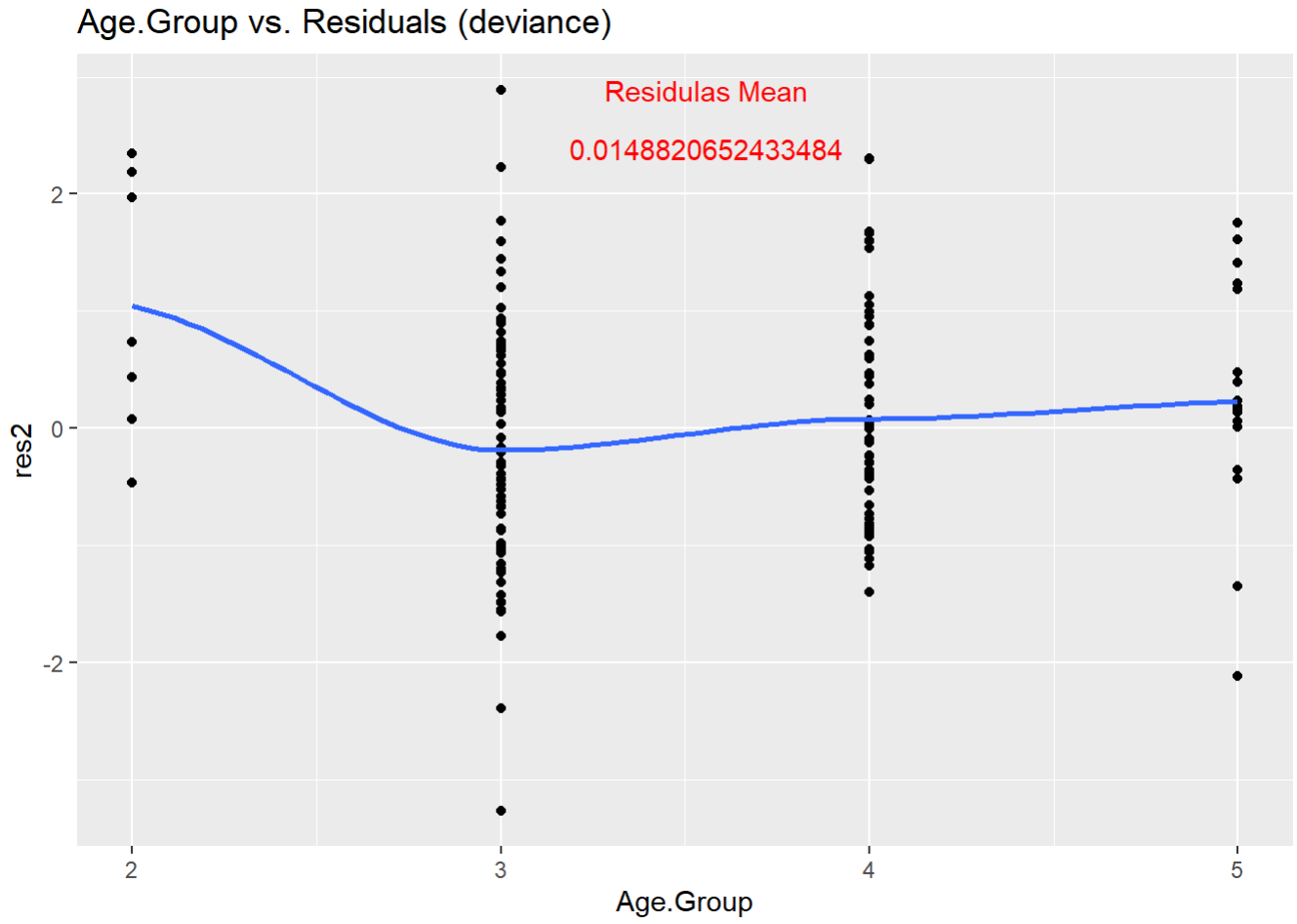
```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1.985
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 2.015
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 2.3189e-016
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 4.0602
```



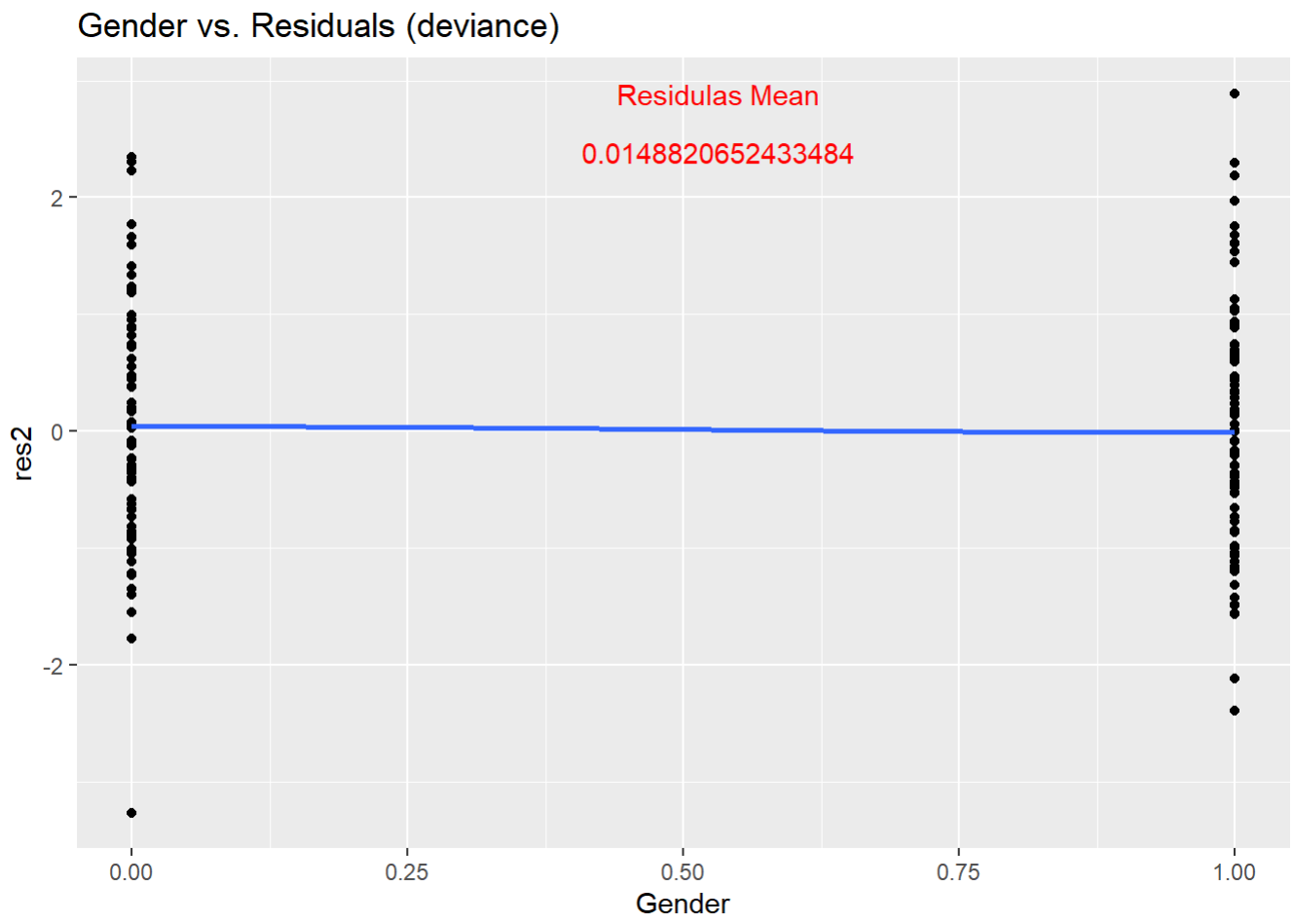

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.005
```

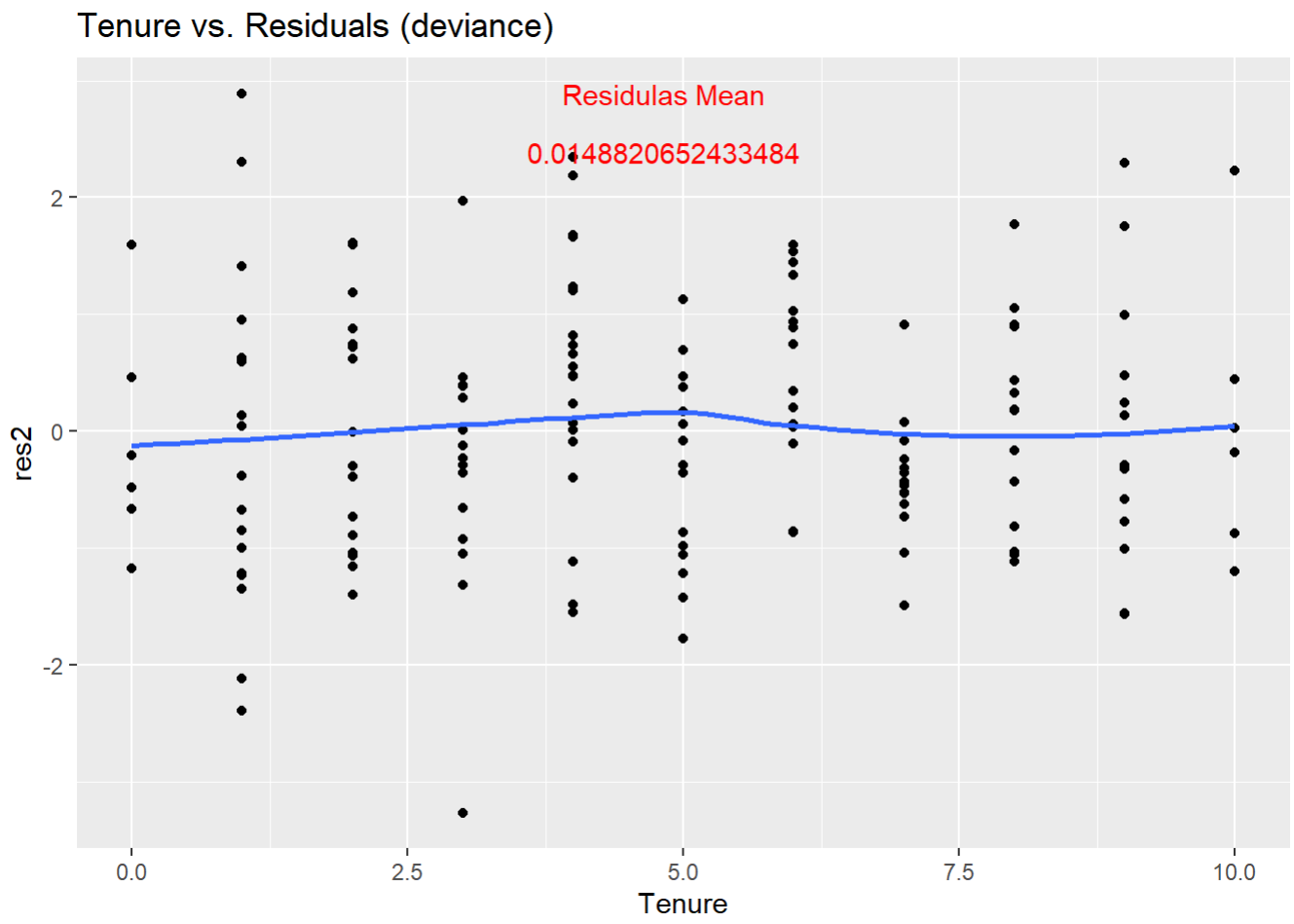
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```




```
## `geom_smooth()` using formula 'y ~ x'
```



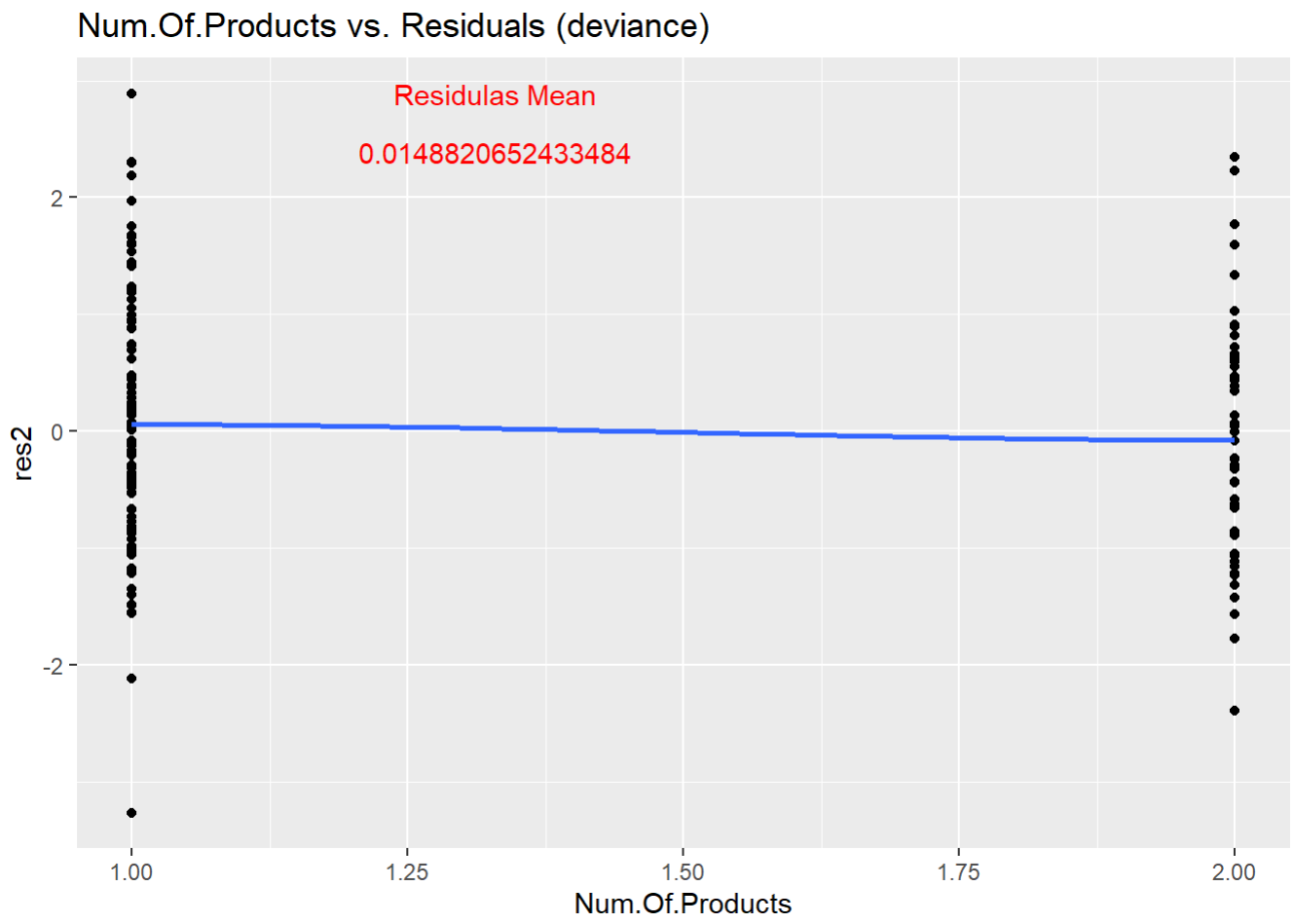

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.995
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```



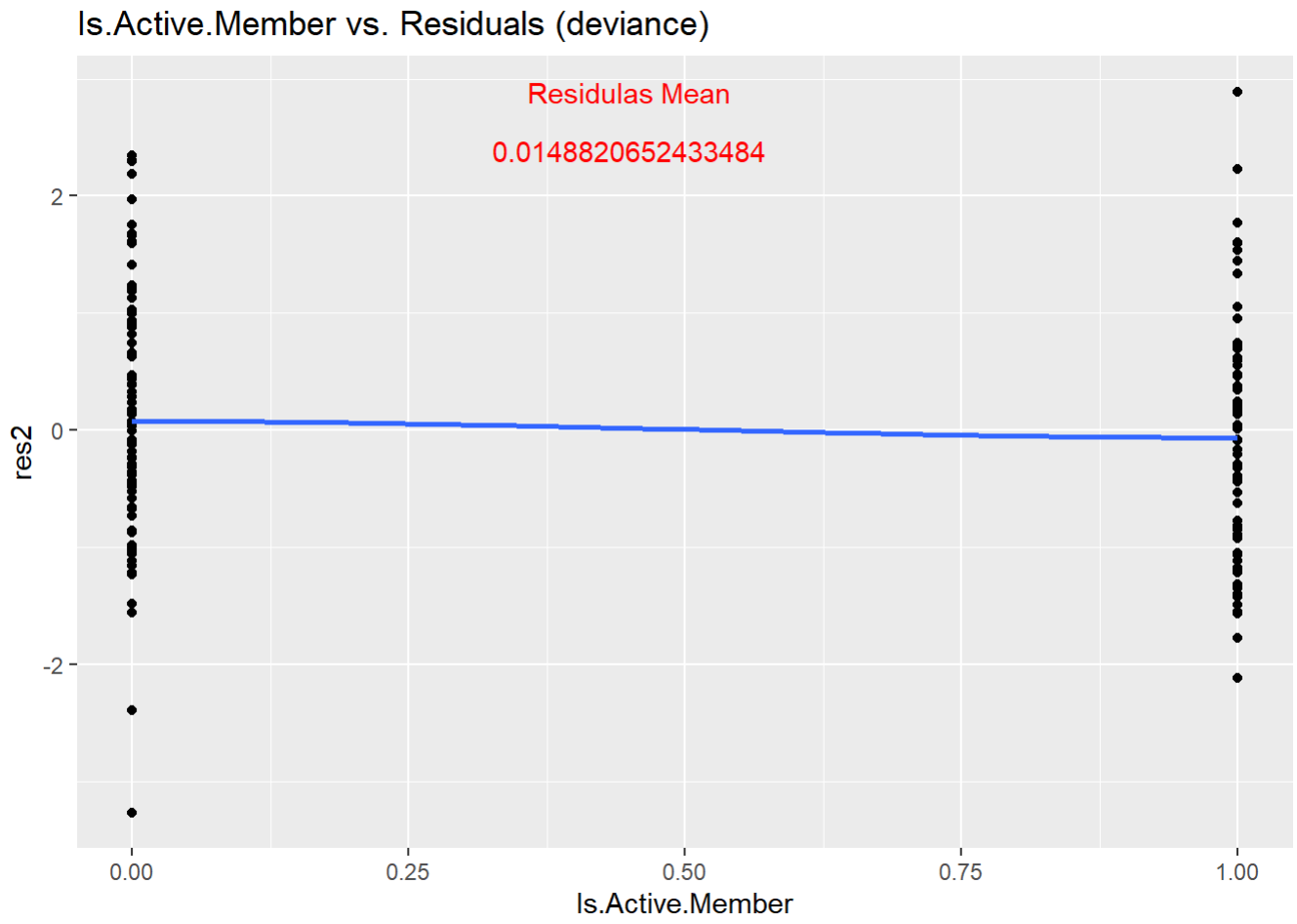

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.005
```

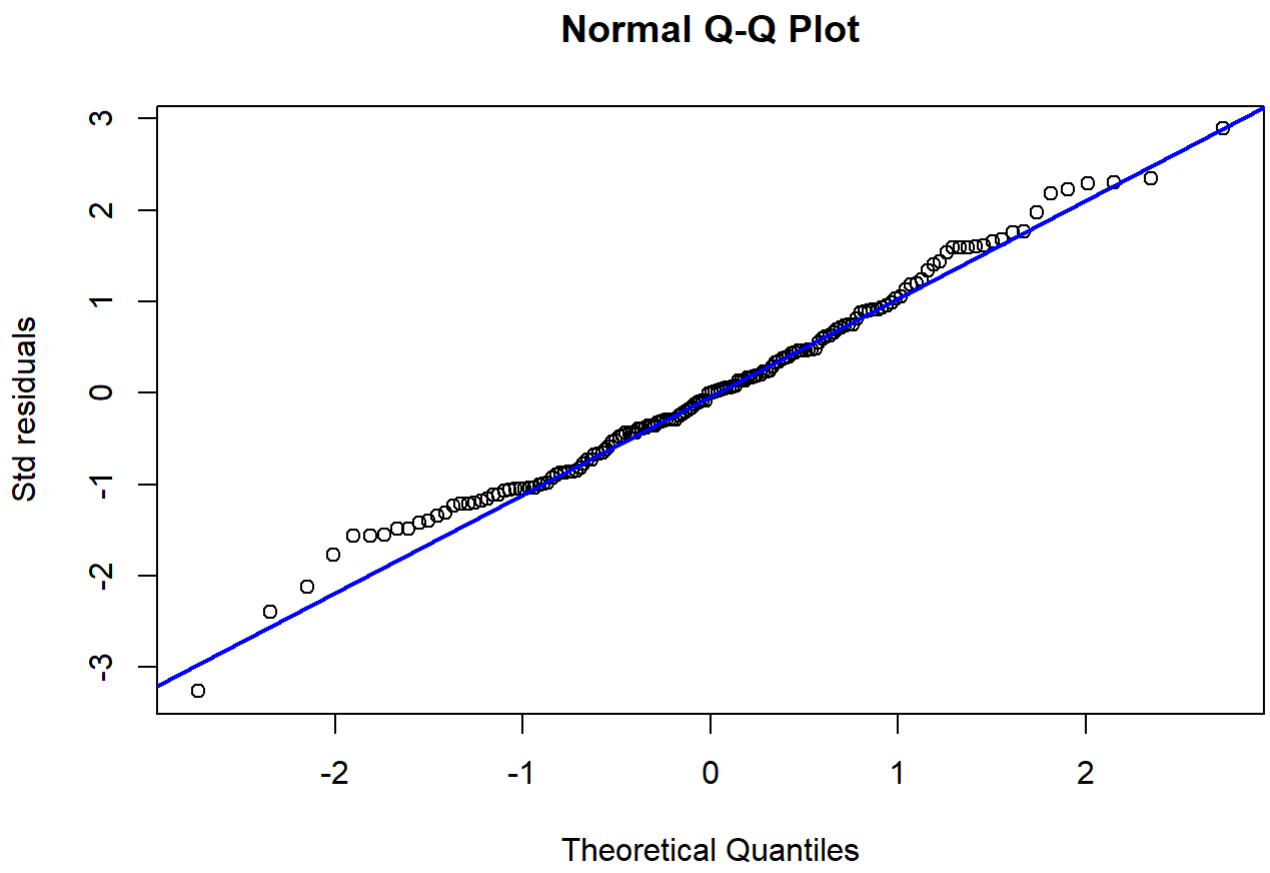
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

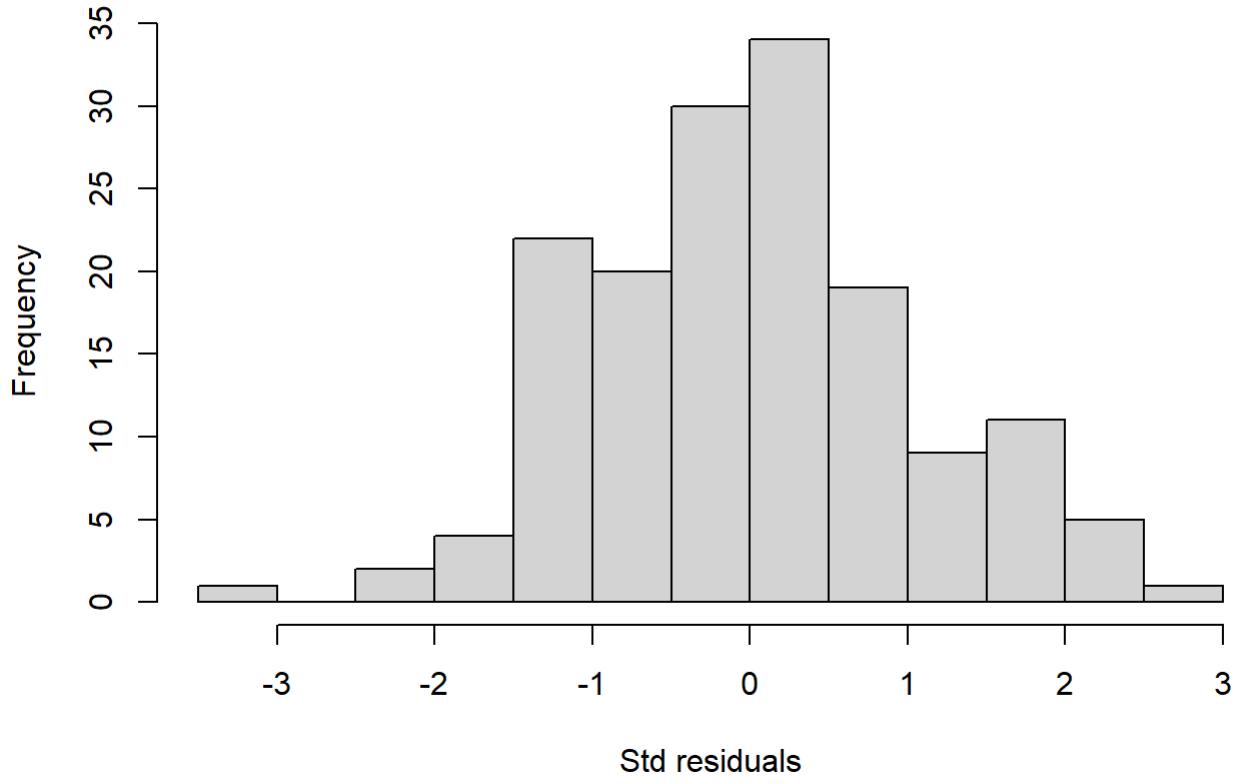
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```




```
# Residuals Normality
qqnorm(res2, ylab="Std residuals")
qqline(res2,col="blue",lwd=2)
```




```
hist(res2,xlab="Std residuals", main="")
```




```
# Hypothesis testing for Normality of residuals
# Null hypothesis is the data is normal distributed. Want LARGE p-value in order NOT to reject.
shapiro.test(res2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res2
## W = 0.99036, p-value = 0.3583
```

Goodness of fit - Dispersion parameter

```
# Dispersion parameter
D = deviance(model2)
DF = model2$df.residual
phi = D/ DF
print("Dispersion parameter:")
```

```
## [1] "Dispersion parameter:"
```

```
phi
```

```
## [1] 1.131172
```

Answer

The goodness of fit has been improved compared to model1. Looking at the p-values of 0.1282109 and 0.200838 of the hypotheses testing using both the deviances and Pearson residuals, respectively, we cannot reject the null hypothesis that model2 fits the data well, hence we conclude the model does fit the data well. The dispersion parameter, ϕ, equals to 1.131172 and is smaller than model1's parameter and also suggesting model2 is not overdispersed since $\phi < 2$. The linearity assumptions holds for the most part as well as the independence assumption as seen by the plots of the residuals against each predictor where the predictors are uncorrelated with the residuals. The mean of model2's residuals is closer to 0 than the mean of model1's residuals. The normality of the residuals has been improved compared to model1 as seen by the Q-Q Normal and Histogram plots. Also, the Shapiro-Wilk test for normality indicate the residuals are normally distributed. The null hypothesis of the Shapiro-Wilk test is that the data is normal distributed and with a high p-value of 0.3583, we cannot reject the null and conclude the residuals are normally distributed.

(e) 2.5 pts - Overall, would you say model2 is a good-fitting model? If so, why? If not, what would you suggest to improve the fit and why? Note, we are not asking you to spend hours finding the best possible model but to offer plausible suggestions along with your reasoning.

```
model3 = glm(Staying ~ Age.Group+Gender+Num.Of.Products+Is.Active.Member, weights = Employees, d
ata = data, family = binomial)
summary(model3)
```

```
##
## Call:
## glm(formula = Staying ~ Age.Group + Gender + Num.Of.Products +
##     Is.Active.Member, family = binomial, data = data, weights = Employees)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2477  -0.7925   0.0032   0.6832   2.9201
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.92617    0.31341  -6.146 7.96e-10 ***
## Age.Group         1.23011    0.07499  16.404  < 2e-16 ***
## Gender           -0.55111    0.09313  -5.918 3.26e-09 ***
## Num.Of.Products  -1.42819    0.11115 -12.849  < 2e-16 ***
## Is.Active.Member -0.86947    0.09458  -9.193  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 981.04  on 157  degrees of freedom
## Residual deviance: 171.99  on 153  degrees of freedom
## AIC: 602.71
##
## Number of Fisher Scoring iterations: 4
```

```
## Test for GOF: Using deviance residuals
print("Test for GOF: Using deviance residuals. chi_statistic and p-value:")
```

```
## [1] "Test for GOF: Using deviance residuals. chi_statistic and p-value:"
```

```
c(deviance(model3), 1-pchisq(deviance(model3),model3$df.residual))
```

```
## [1] 171.9852777   0.1398126
```

```
## Test for GOF: Using Person residuals
pearres = residuals(model3,type="pearson")
pearson.tvalue = sum(pearres^2)
print("Test for GOF: Using Person residuals. chi_statistic and p-value:")
```

```
## [1] "Test for GOF: Using Person residuals. chi_statistic and p-value:"
```

```
c(pearson.tvalue, 1-pchisq(pearson.tvalue,model3$df.residual))
```

```
## [1] 166.3654975   0.2174811
```

```
################################################################################
# Linearity assumption
odds = data$Staying / (1 - data$Staying) # Calculating odds. "Staying" is the probability and we
need to plot the log(odds) vs. Predictor to asses Linearity assumption.
Age.Group = data$Age.Group
Gender = data$Gender
#Tenure = data$Tenure
Num.Of.Products = data$Num.Of.Products
Is.Active.Member = data$Is.Active.Member
odds_df3 = data.frame(Age.Group, Gender, Num.Of.Products, Is.Active.Member, odds)

predictors = c("Age.Group", "Gender", "Num.Of.Products", "Is.Active.Member")
n = length(predictors)

for (i in c(1:n)){
  print(ggplot(odds_df3, aes(x=odds_df3[,i], y=log(odds))) + ggtitle(paste(predictors[i], "vs. L
og(odds)")) + xlab(predictors[i]) + geom_point() +
    scale_colour_hue(l=50) + # Use a slightly darker palette than normal
    geom_smooth(method="loess",   # Add regression line
                se=FALSE,    # Don't add shaded confidence region
                fullrange=TRUE)) # Extend regression lines
}
```

```
## `geom_smooth()` using formula 'y ~ x'
```

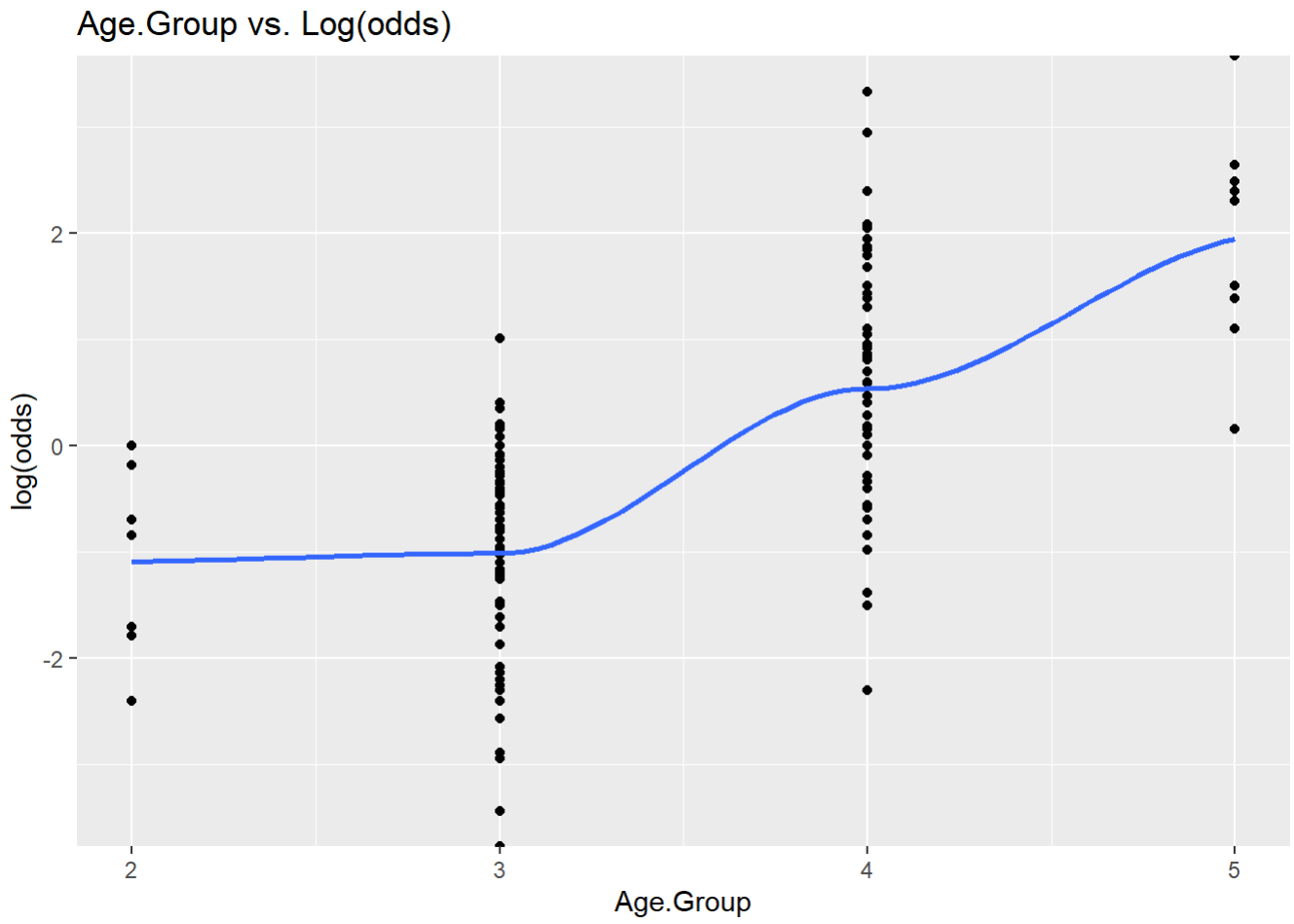
```
## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1.985
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 2.015
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 5.0803e-016
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 4.0602
```




```
## `geom_smooth()` using formula 'y ~ x'
```

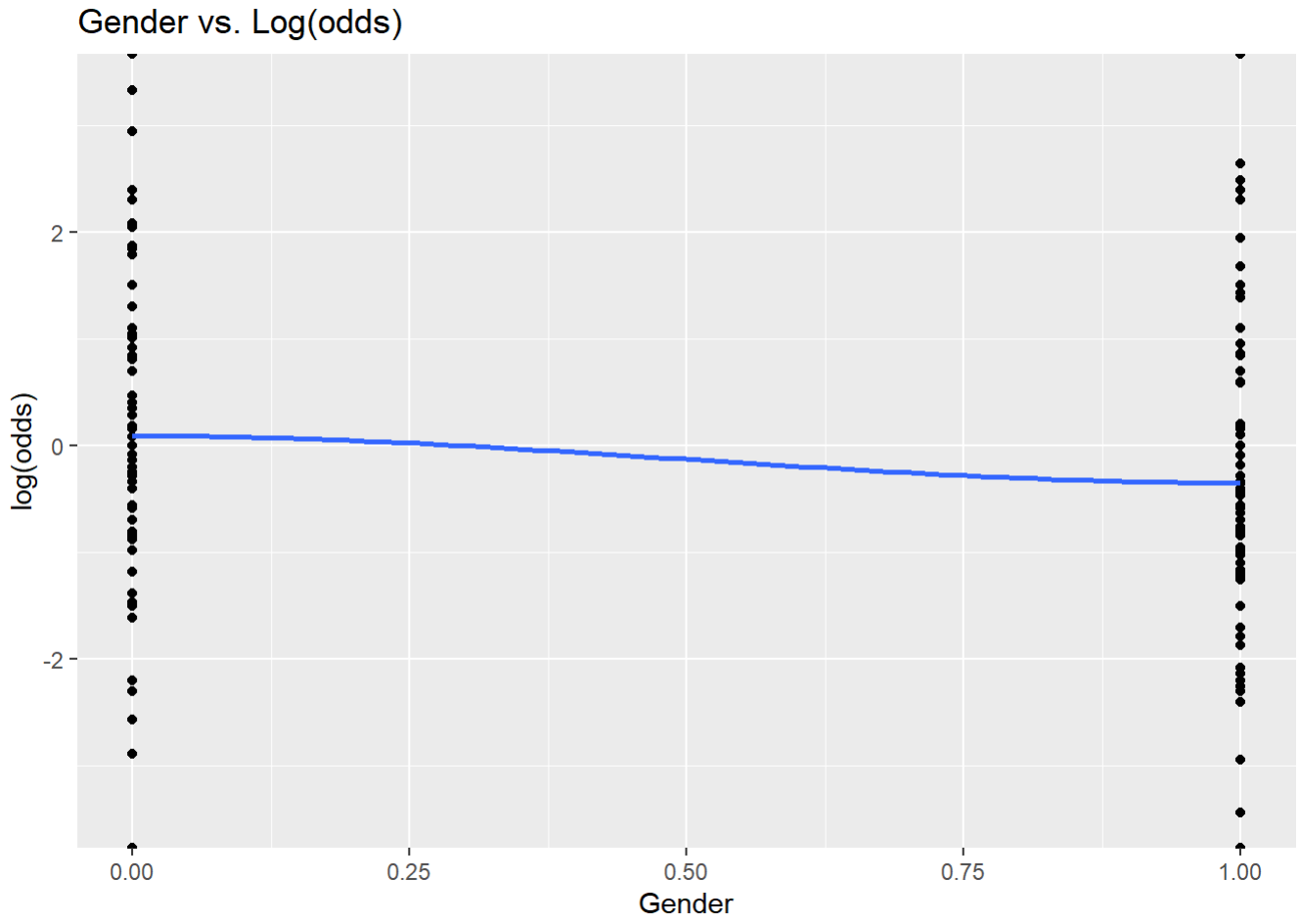
```
## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```




```
## `geom_smooth()` using formula 'y ~ x'
```

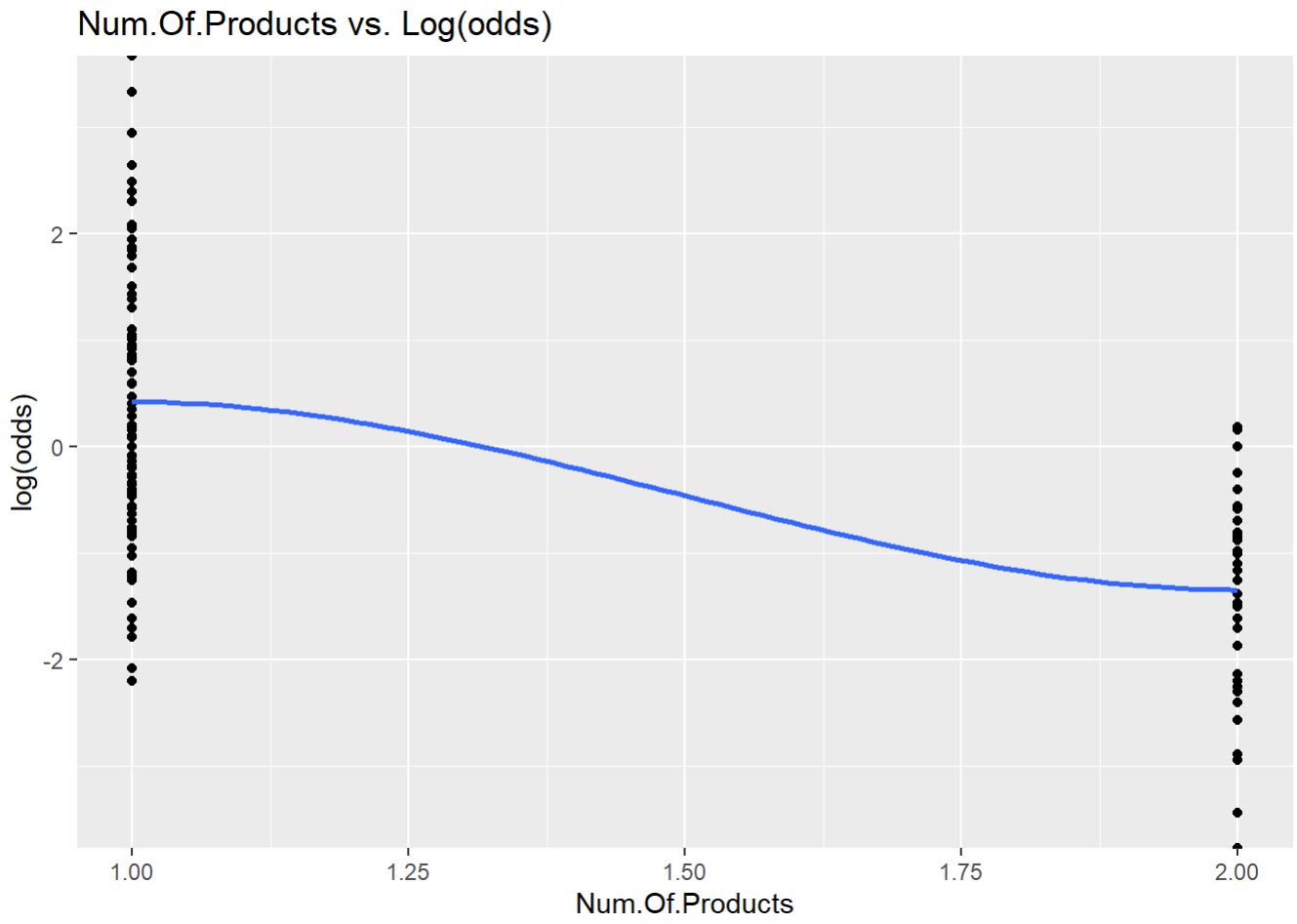
```
## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.995
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```




```
## `geom_smooth()` using formula 'y ~ x'
```

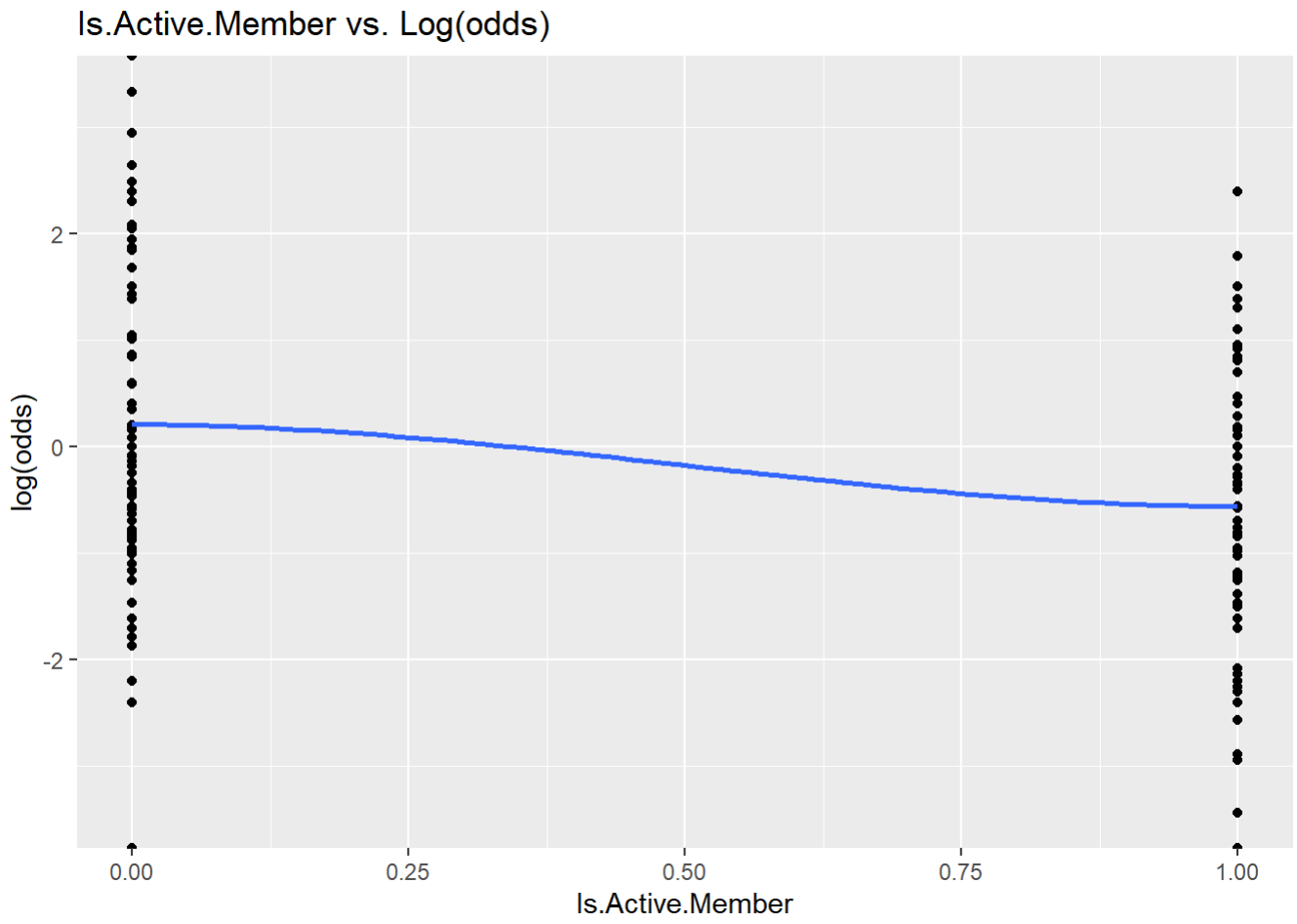
```
## Warning: Removed 10 rows containing non-finite values (stat_smooth).
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 7.1561e-031
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```




```
##############################################################################################
# Independence assumption
res3 = resid(model3,type="deviance")

res_df3 = data.frame(Age.Group, Gender, Num.Of.Products, Is.Active.Member, res3)
for (i in c(1:n)){
print(ggplot(res_df3, aes(x=res_df3[,i], y=res3)) + ggtitle(paste(predictors[i], "vs. Residuals
 (deviance)")) + xlab(predictors[i]) + geom_point() +
  annotate("text", x= mean(res_df3[,i]), y=max(res3)-0.5, label= mean(res3), col="red") +
  annotate("text", x= mean(res_df3[,i]), y=max(res3), label= "Residulas Mean", col="red") +
  scale_colour_hue(l=50) + # Use a slightly darker palette than normal
  geom_smooth(method="loess",   # Add regression line
              se=FALSE,    # Don't add shaded confidence region
              fullrange=TRUE)) # Extend regression lines
}
```

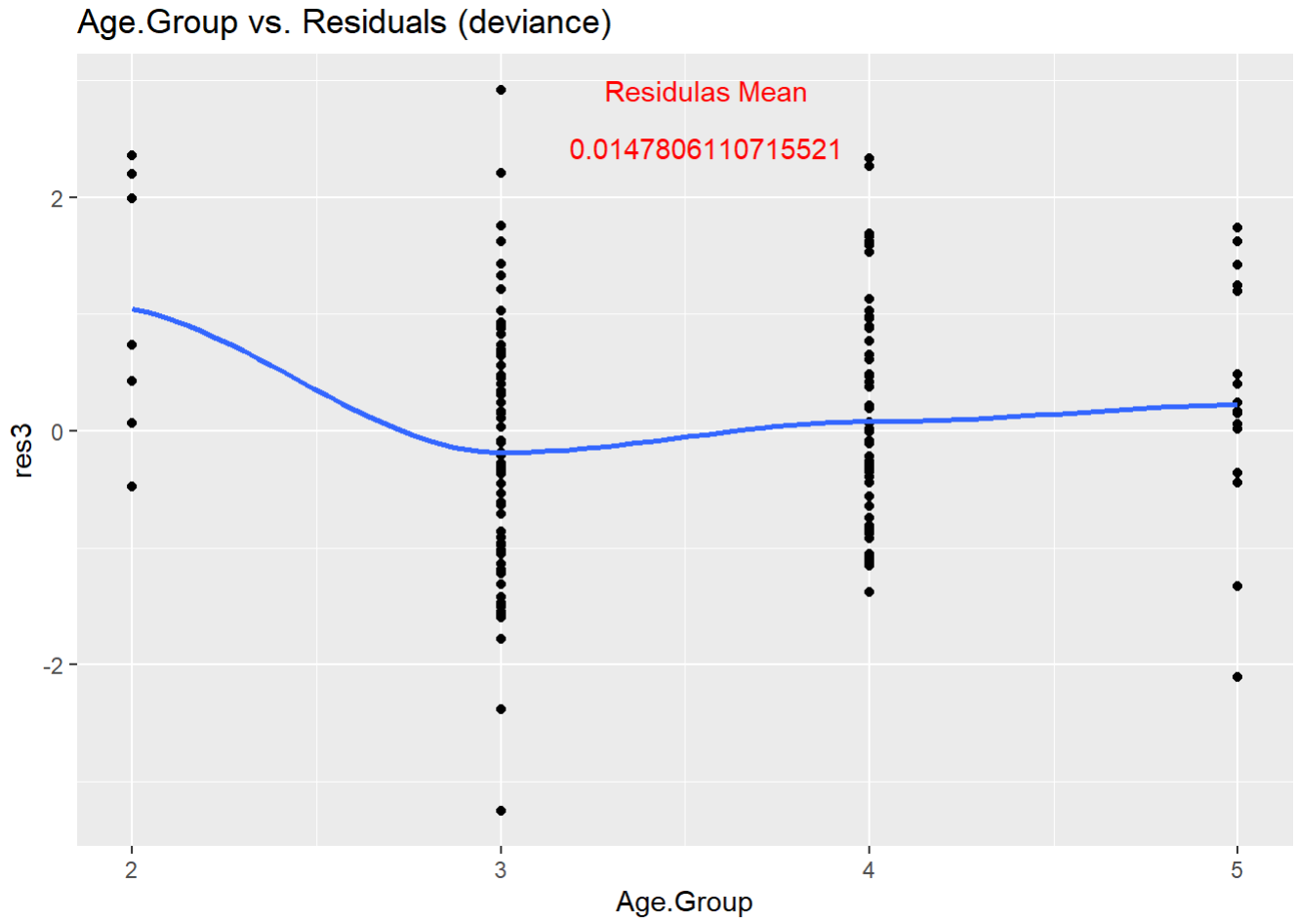
```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1.985
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 2.015
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 2.3189e-016
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 4.0602
```



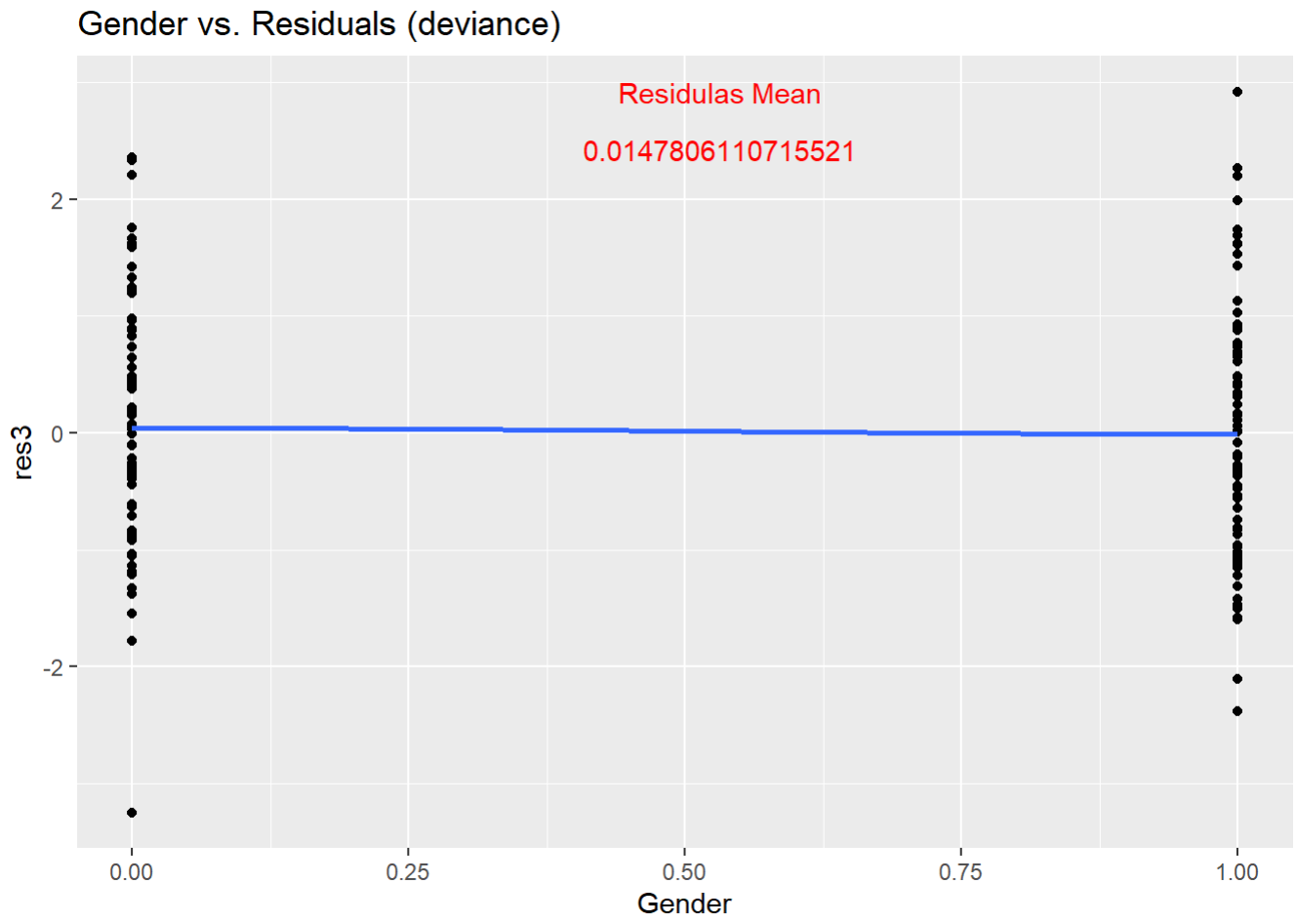

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```



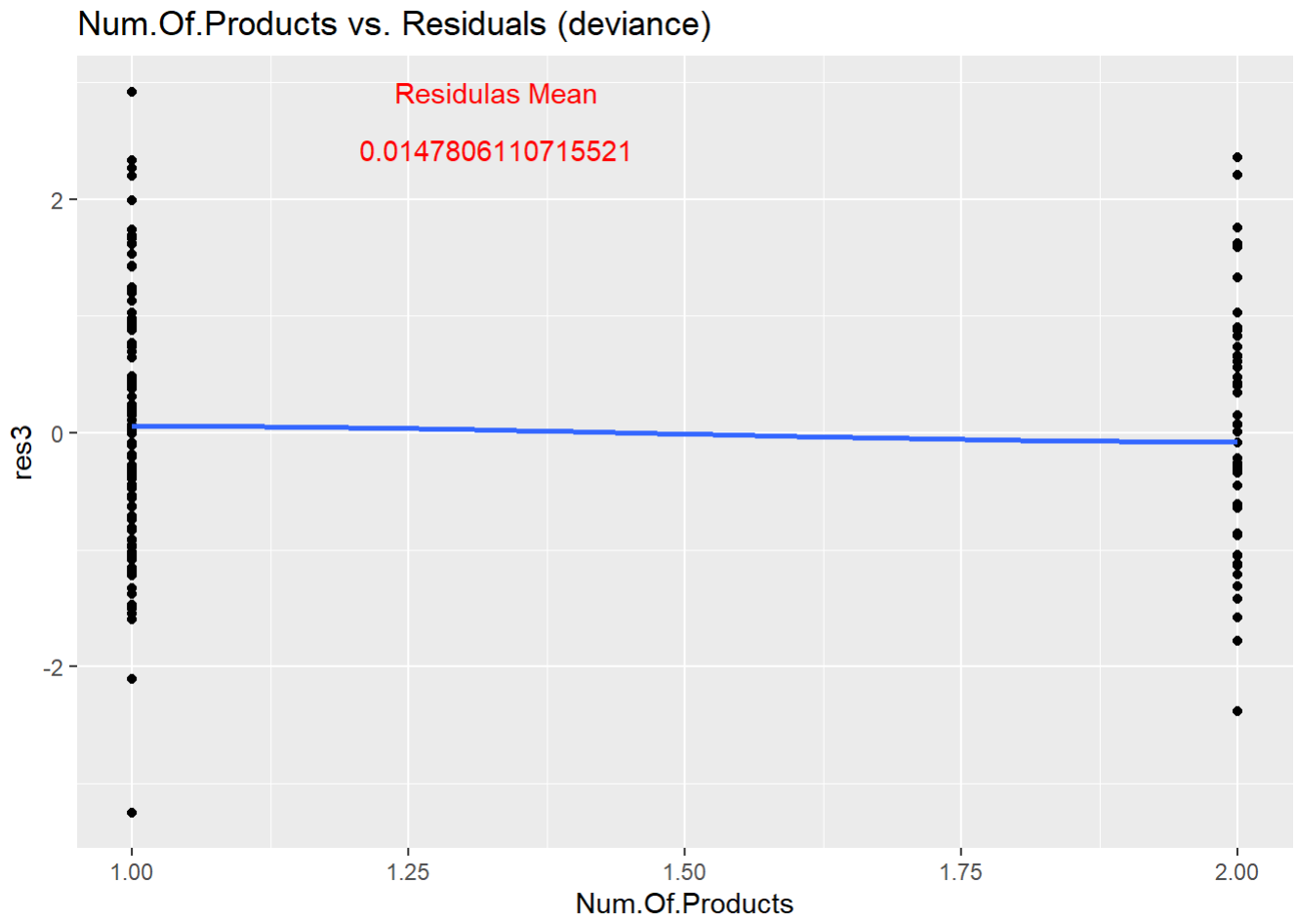

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.995
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```



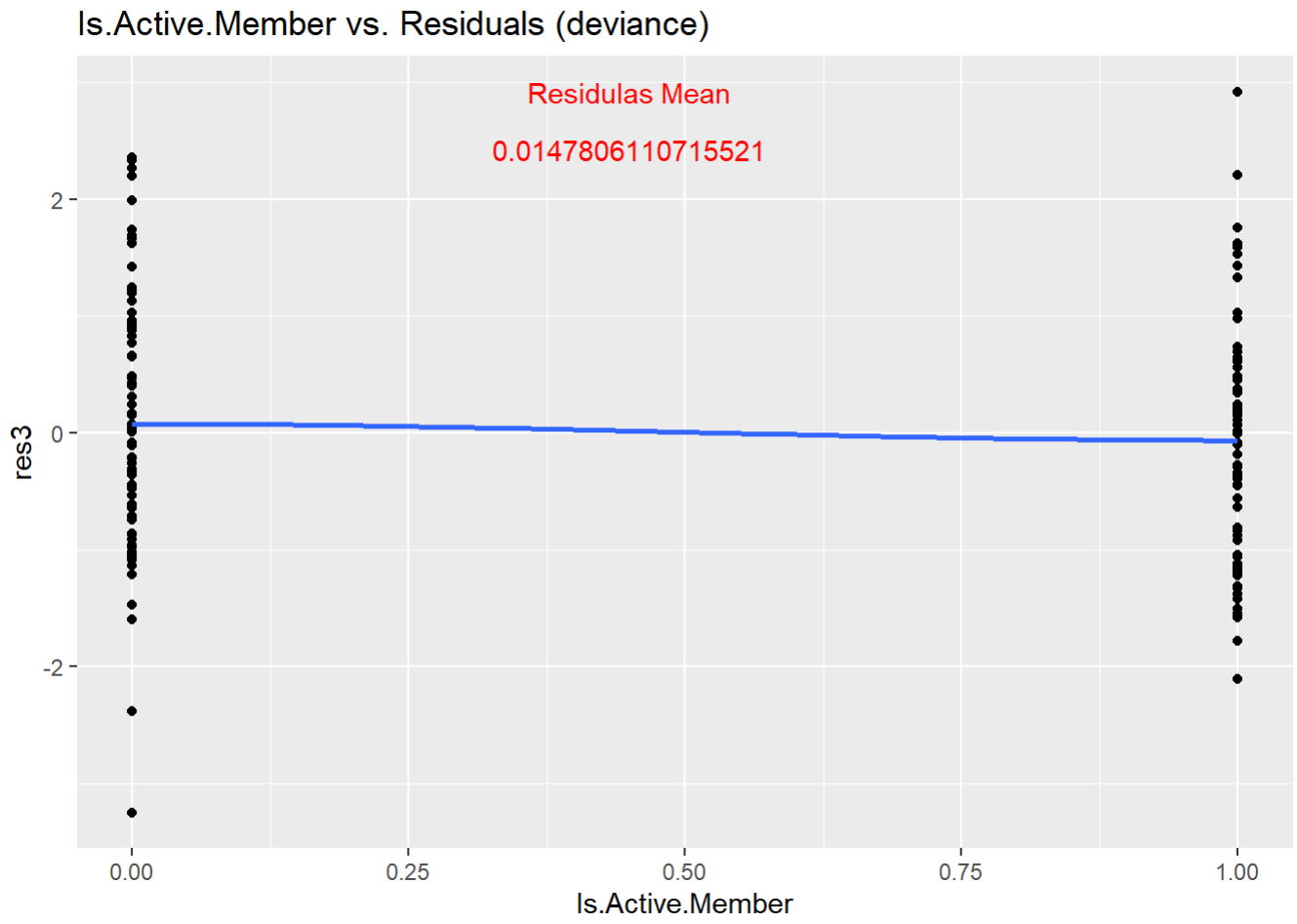

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at -0.005
```

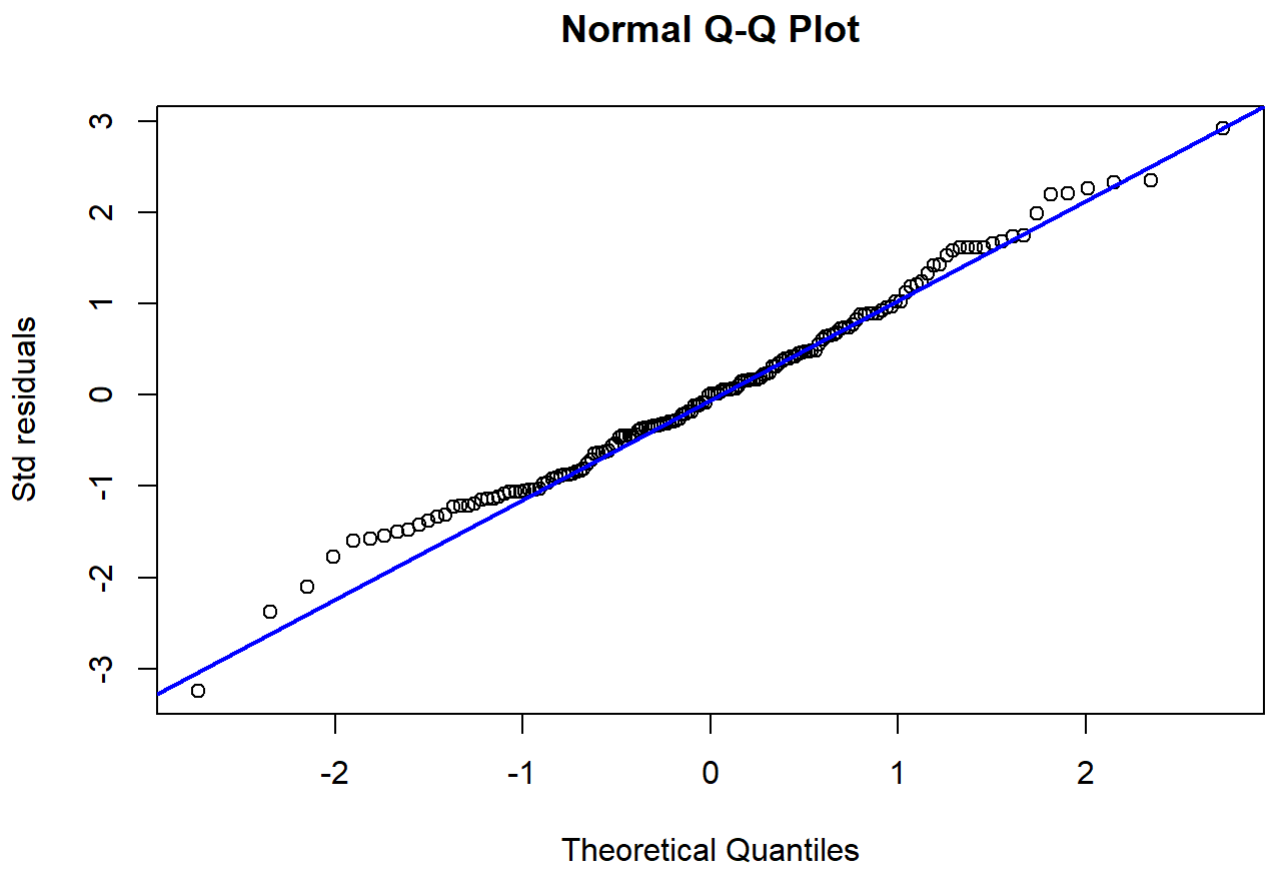
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.005
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

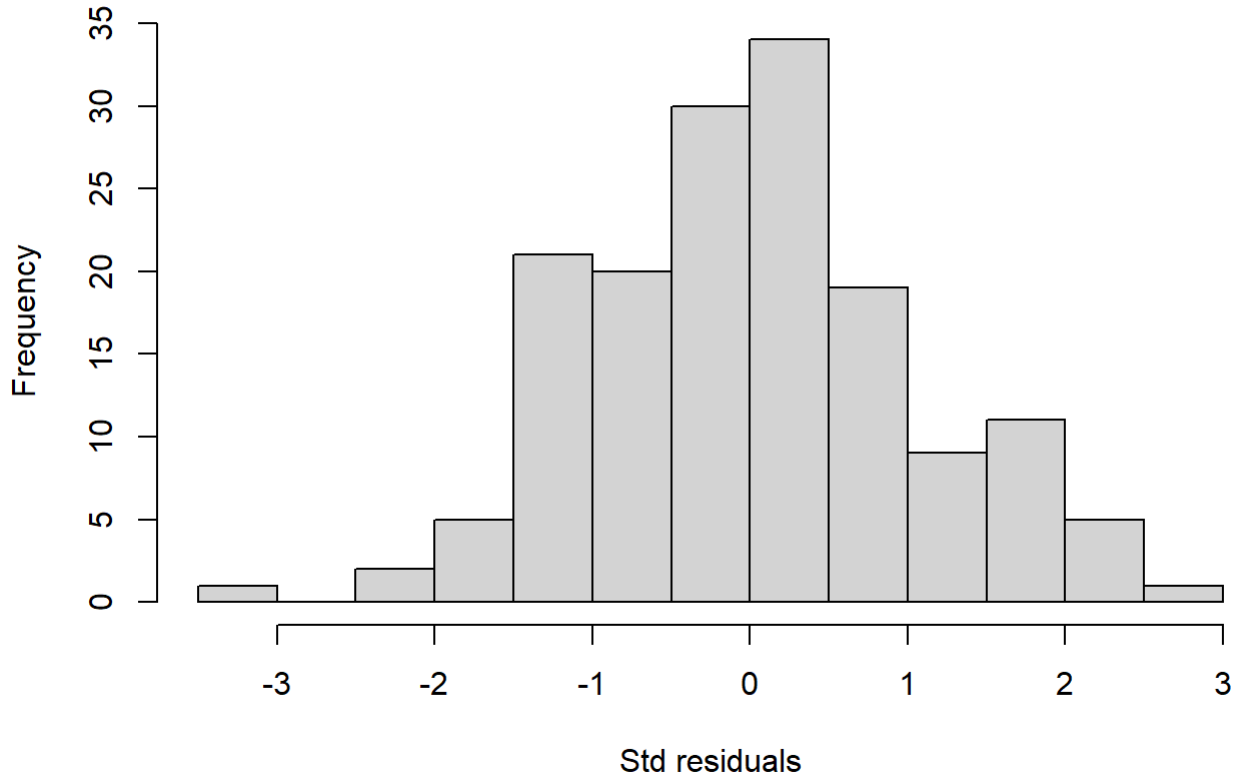
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.01
```




```
qqnorm(res3, ylab="Std residuals")
qqline(res3,col="blue",lwd=2)
```




```
hist(res3,xlab="Std residuals", main="")
```




```
# Hypothesis testing for Normality of residuals
# Null hypothesis is the data is normal distributed. Want LARGE p-value in order NOT to reject.
shapiro.test(res3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res3
## W = 0.99036, p-value = 0.3581
```

```
# Dispersion parameter
D = deviance(model3)
DF = model3$df.residual
phi = D/ DF
print("Dispersion parameter:")
```

```
## [1] "Dispersion parameter:"
```

```
phi
```

```
## [1] 1.124087
```

Answer

Overall I would say model2 is a goof fitting model, because as seen above, based on the hypothesis testing we concluded the model does fit the data and the residuals plots show they are, for the most part, normally distributed. The Shapiro-Wilk test also indicates the residuals are normaly distributed. However, looking at the plots above of model3, including only $Age.Group$, $Gender$, $Num.Of.Products$, and $Is.Active.Member$, we can see a slightly better fitting as indicated by a slightly lower ϕ dispersion parameter. Model3's p-values of the hypotheses testing using both the deviances and Pearson residuals are higher than model2's p-values which is also a good indication.

Question 5: Prediction - 6 pts

Suppose there is an employee with the following characteristics:

1. **Age.Group**: 2
2. **Gender**: 0
3. **Tenure**: 2
4. **Num.Of.Products**: 2
5. **Is.Active.Member**: 1

(a) 2 pts - Predict their probability of staying using model1.

```
employee = data.frame(Age.Group=2, Gender=0, Tenure=2, Num.Of.Products=2, Is.Active.Member=1)
predict(model1, employee, type="response")
```

```
##         1
## 0.1997319
```

Answer

Using model1, the probability of staying given the predicting variables for employee above is 0.1997319 or 19.97%. Also see manual calculation:

$odds$ = $e^{2.1457-1.7668*Num.Of.Products}$

$odds_{employee}$ = $e^{2.1457-1.7668*2}$ = 0.249598912

$p(Staying|employee)$ = $odds_{employee}$ / (1 + $odds_{employee}$) = 0.249598912 / (1 + 0.249598912) = ***0.19974***

(b) 2 pts - Predict their probability of staying using model2.

```
predict(model2, employee, type="response")
```

```
##         1
## 0.03987005
```

Answer

Using model2, the probability of staying given the predicting variables for employee above is 0.03987005 or 3.987%. Also see manual calculation:

$odds$ = $e^{-1.903330+1.229014*Age.Group-0.551438*Gender-0.003574*Tenure-1.428767*Num.Of.Products-0.871460*Is.Active.Member}$

$odds_{employee}$ = $e^{-1.903330+1.229014*2-0.551438*0-0.003574*2-1.428767*2-0.871460*1}$ = 0.041525648

$p(Staying|employee)$ = $odds_{employee}$ / (1 + $odds_{employee}$) = 0.041525648 / (1 + 0.041525648) = ***0.03987***

(c) 2 pts - Comment on how your predictions compare.

Answer

We can see that model1 and model2 give very different probabilities of Staying given $employee$. model1 predicted 19.97% of staying while model2 predicted 3.987%. These findings suggest we should be careful when fitting and predicting a logistic regression model. More specifically, we should always perform and evaluate goodness of fit and testing for different subset of coefficient to find the best possible model.

Thank You!