Chapter 10: Logistic regression

Contents

1 Introduction

Example *Political party*

Consider the *political_party.xlsx* file. Import this excel file in R as `political_party`. In this data set, one of the variables is the variable *Republican* which indicates whether a person votes for the Republican Party or not. We have 283 respondents ($n = 283$).

Variable	Type	Description
`political_party`	nominal	1: Republican 2: Democrat 3: Independent
`Republican`	response	based on the variable `political_party` 0: Not Republican 1: Republican
`gender`	indicator	0: Female 1: Male
`pro_capital_punishment`	continuous	10 point scale, higher values indicating greater support for the position.
`pro_welfare_reform`	continuous	10 point scale, higher values indicating greater support for the position.
`pro_fed_support_ed` (Federal support of education)	continuous	10 point scale, higher values indicating greater support for the position.

We always want to estimate the probability of $response = 1$ **(here:** $P[Republican = 1]$**).** The category with value 1 is called the "target category". We will start with univariate logistic regression with as explanatory variable `pro_capital_punishment`.

2 Regression model with binary response variable

- Consider the regression model $Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ with $Y_i \in \{0, 1\}$. Then, $E(Y) = \beta_0 + \beta_1 x$.
- Assuming that Y_i is a Bernoulli distributed random variable, the following table holds:

Y_i	Probability
1	$P(Y_i = 1) = p$
0	$P(Y_i = 0) = 1 - p$

We can show that $E(Y) = 1 \cdot p + 0 \cdot (1 - p) = p$

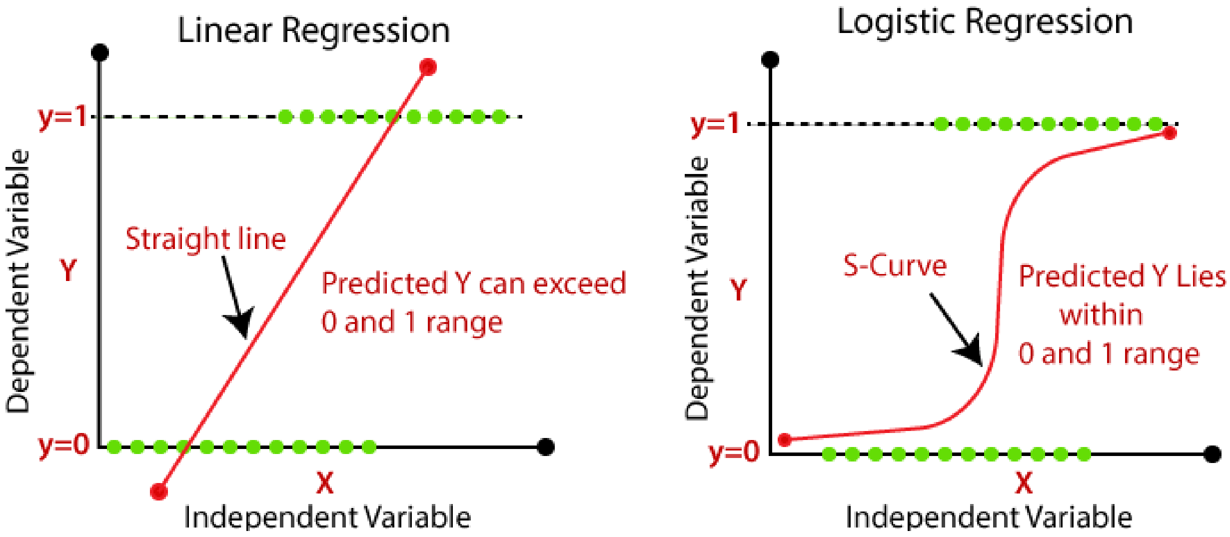
$\rightarrow$ Combining these results gives: $E(Y) = \beta_0 + \beta_1 x = p$

Interpretation:
The average response is the probability that $Y = 1$.

Problem:
Restriction on the response function:
$0 \leq E(Y) = p \leq 1$

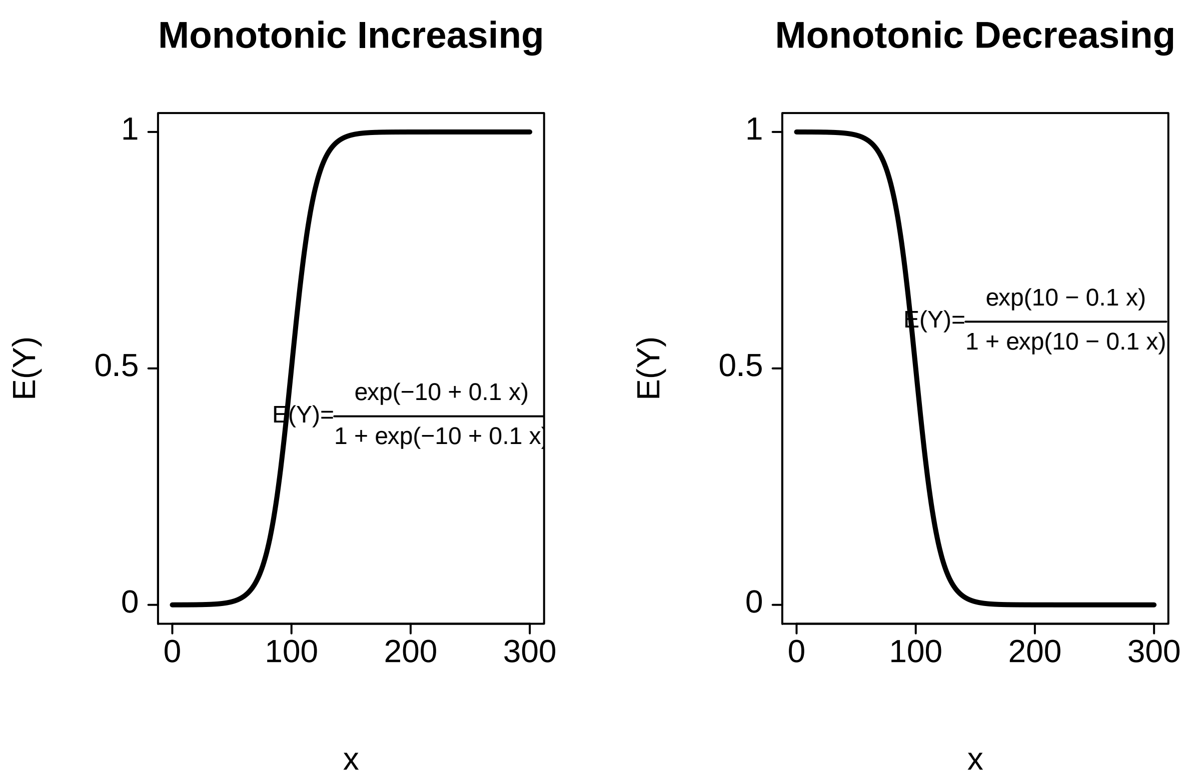
$\Rightarrow$ a linear response function is not possible!! Linear regression is used to predict a continuous dependent variable using a given set of independent variables.

Logistic Regression is used to predict a binary (0 or 1) dependent variable using a given set of independent variables.



3 Simple logistic regression

3.1 Logistic response function



The logistic response function has the form:
$p = E(Y) = \frac{\exp(\beta_0+\beta_1 x)}{1+\exp(\beta_0+\beta_1 x)}$
or
$p = E(Y) = \frac{1}{1+\exp(-\beta_0-\beta_1 x)}$

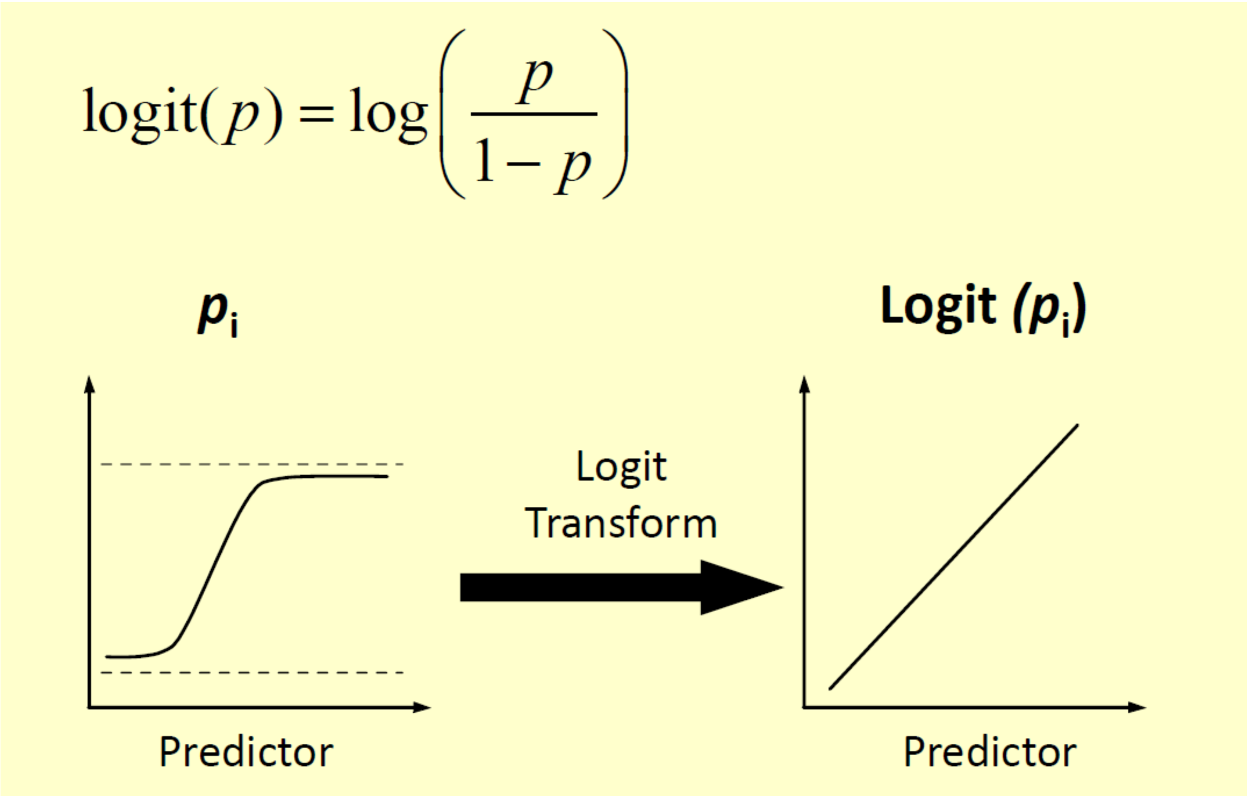
It can be seen that the relationship between the probability $p(= P[Y = 1])$ and the independent variable x is represented by a logistic curve. Note that this relationship is nonlinear.

3.2 Properties of logistic response function

Some properties of the logistic response function:

- Either monotone increasing or monotone decreasing (depending on sign of β_1).
- Is almost linear in the range where $E(Y)$ ranges from 0.2 to 0.8.
- It approaches 0 and 1 at the two ends of the x range.
- **It can be linearized:** The logistic response function can be transformed to a linear one: Using the LOGIT transformation (i.e., $p' = \ln(\frac{p}{1-p})$), we obtain:
  $p' = \beta_0 + \beta_1 x$
  with $p = P(Y = 1)$, p' the logit mean response and $\frac{p}{1-p}$ the odds.

$\ln(\frac{p}{1-p}) = \beta_0 + \beta_1 x$
is called the Logit or Log(odds).



3.3 Interpretation of the odds

If the probability of an event is p, then the odds O of the event is
$O = \frac{p}{1-p} = \frac{probability\ of\ event}{probability\ of\ no\ event}$

- The odds for winning the lottery is the probability of winning the lottery divided by the probability of not winning the lottery.

- The odds of having a Facebook account is the probability of having a Facebook account divided by the probability of not having a Facebook account.

An odds of 4 means that the expected number of events is four times the number of no events.

Probability p	Odds O
0.1	0.11
0.2	0.25
0.3	0.43
0.4	0.67
0.5	1
0.6	1.5
0.7	2.33
0.8	4
0.9	9

$Odds < 1$ corresponds with $p < 0.5$.
Odds do have a lower bound of 0, but there is no upper bound.

Once you have *odds*, you can derive the probability of the event by
$p = \frac{odds}{1+odds}$

3.4 Assessing the model: the log-likelihood statistic

We state the simple logistic regression model as
$p = E(Y) = \frac{\exp(\beta_0+\beta_1 x)}{1+\exp(\beta_0+\beta_1 x)}$
with $p = P(Y = 1)$ and x the explanatory variable.

Remark:

- In regression analysis, *method of least squares* was used to obtain parameter estimates.
- In logistic regression, *maximum likelihood estimation* is used to obtain parameter estimates.

3.4.1 Log-likelihood function

Y_i are independent Bernoulli random variables with $P(Y_i = 1) = p_i$.

The probability function is: $f_i(Y_i) = p_i^{Y_i}(1-p_i)^{1-Y_i}$ where $Y_i \in \{0,1\}$ (i.e., Y_i can only take the values 0 and 1).

Joint probability function: $g(Y_1, Y_2, ...Y_n) = \prod_{i=1}^n f_i(Y_i) = \prod_{i=1}^n p_i^{Y_i}(1-p_i)^{(1-Y_i)}$

Taking the natural logarithm:
$\log_e(g(Y_1, Y_2, ..., Y_n)) = \log_e\left(\prod_{i=1}^n p_i^{Y_i}(1-p_i)^{(1-Y_i)}\right)$

Or the **log-likelihood** can be written as:
$\log_e(g(Y_1, Y_2, ..., Y_n)) = \sum_{i=1}^n \left[Y_i \log_e\left(\frac{p_i}{1-p_i}\right)\right] + \sum_{i=1}^n \log_e(1-p_i)$

Remark:
$p_i = \frac{\exp(\beta_0+\beta_1 x_i)}{1+\exp(\beta_0+\beta_1 x_i)}$
and
$1 - p_i = \frac{1}{1+\exp(\beta_0+\beta_1 x_i)}$

3.4.2 Maximum likelihood estimates

- Chose those estimates for β_0 and β_1 which maximizes the log-likelihood.

- Find maximum likelihood estimates for β_0 and β_1: b_0 and b_1.
- Substitute these into the response function to obtain *fitted response function* $\hat{p}$: $\hat{p}_i = \frac{\exp(b_0+b_1x_i)}{1+\exp(b_0+b_1x_i)}$
- Use the logit transformation to obtain *fitted logit response function* $\hat{p}' = b_0 + b_1x$ with $\hat{p}' = \log_e\left(\frac{\hat{p}}{1-\hat{p}}\right) = b_0 + b_1x = \log(o\hat{d}ds)$

3.5 How to obtain parameter estimates in R

A maximum likelihood estimation procedure can be used to obtain the parameter estimates. Since no analogical procedure exists, an iterative procedure is employed to obtain these estimates.

Example *Political party*

```
PP <- political_party
names(PP)
```

```
## [1] "subid"                   "political_party"        "gender"
## [4] "pro_capital_punishment" "pro_welfare_reform"     "pro_fed_support_ed"
## [7] "Republican"
```

```
head(PP)
```

```
## # A tibble: 6 x 7
##   subid political_party gender pro_capital_pun~ pro_welfare_ref~
##   <dbl>           <dbl>  <dbl>            <dbl>            <dbl>
## 1     1               2      0                3                6
## 2     2               2      0                2                5
## 3     3               2      0                1                6
## 4     4               2      0                4                7
## 5     5               2      0                4                6
## 6     6               2      0                4                6
## # ... with 2 more variables: pro_fed_support_ed <dbl>, Republican <dbl>
```

```
# Model 1
glm.log1 <- glm(Republican ~ pro_capital_punishment, family = binomial(link = logit), data = PP)
summary(glm.log1)
```

```
##
## Call:
## glm(formula = Republican ~ pro_capital_punishment, family = binomial(link = logit),
##     data = PP)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3090  -0.9461  -0.8183   1.3498   1.6646
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -1.44778    0.38716   -3.74 0.000184 ***
## pro_capital_punishment  0.17520    0.08263    2.12 0.033988 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 361.18  on 282  degrees of freedom
## Residual deviance: 356.62  on 281  degrees of freedom
```

```
## AIC: 360.62
##
## Number of Fisher Scoring iterations: 4
```

The estimated logistic regression function is:
$\hat{p} = P(Republican = 1) = \frac{exp(-1.448+0.175 \cdot ProCapPun)}{1+exp(-1.448+0.175 \cdot ProCapPun)}$

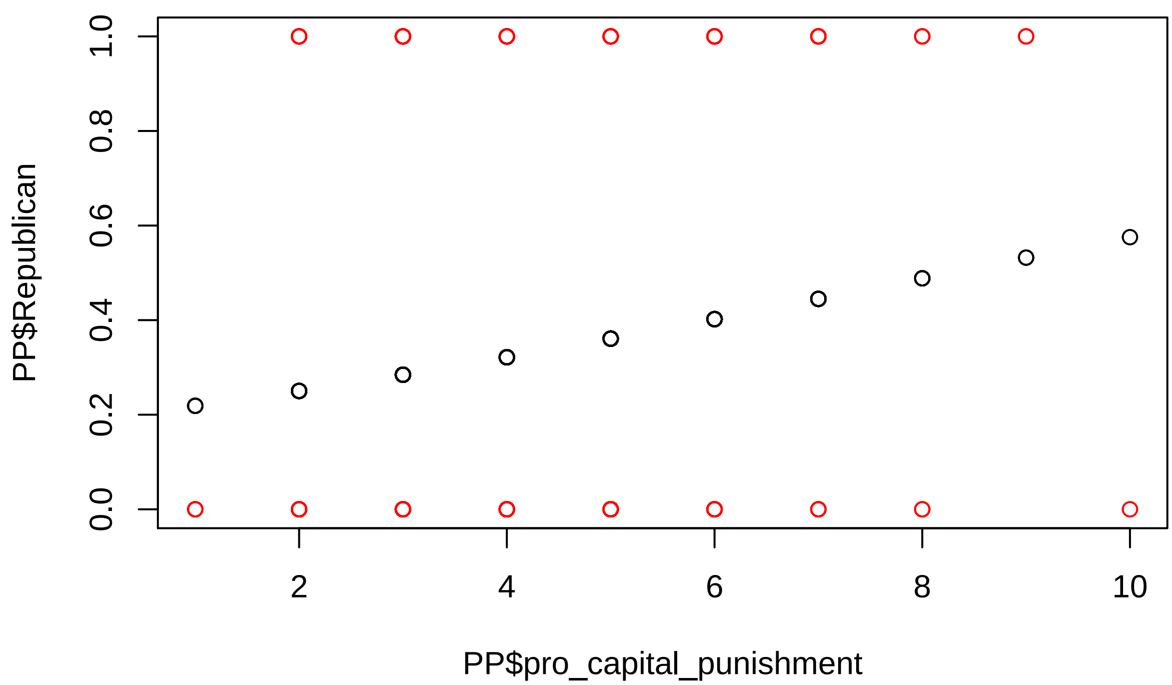
1. Look at predicted values

```r
combine <- data.frame(cbind(PP$pro_capital_punishment, PP$Republican, fitted(glm.log1)))
colnames(combine) <- c("pro_capital_punishment", "Republican", "Fitted value")
head(combine,5)
```

```
##   pro_capital_punishment Republican Fitted value
## 1                      3          0    0.2845133
## 2                      2          0    0.2502308
## 3                      1          0    0.2188158
## 4                      4          0    0.3214787
## 5                      4          0    0.3214787
```

2. This predicted and observed values can be visualized in the following graph.

```r
plot(PP$pro_capital_punishment, PP$Republican, type="p", col="red")
points(PP$pro_capital_punishment, fitted(glm.log1), col="black")
```



3.6 Interpretation of b_1

3.6.1 General

- The interpretation of b_1 is not the same interpretation as the slope in a linear regression model. (= the value of b is there the change in the outcome resulting from a one unit change in the predictor variable)
- The interpretation of b_1 can be: The value of b_1 is the change in the logit of the outcome resulting from a one unit change in the predictor variable.
- We will explain the interpretation by using the concept of *odds*.

Consider the value of the fitted logit response function at $x = x_j$:
$\hat{p}'(x_j) = b_0 + b_1 x_j$

Consider the value of the fitted logit response function at $x = x_j + 1$ (a one unit increase):
$\hat{p}'(x_j + 1) = b_0 + b_1(x_j + 1)$

The difference between the two fitted values:
$\hat{p}'(x_j + 1) - \hat{p}'(x_j) = b_1$

Now, $\hat{p}' = \log_e\left(\frac{\hat{p}}{1-\hat{p}}\right)$ = log of the estimated odds.

b_1 = the difference between the two fitted values:
$b_1 = \hat{p}'(x_j + 1) - \hat{p}'(x_j)$
$b_1 = \log_e(\hat{odds}_{x+1}) - \log_e(\hat{odds}_x)$
$b_1 = \log_e\left(\frac{\hat{odds}_{x+1}}{\hat{odds}_x}\right)$

$\Rightarrow$ Odds ratio $= OR = \frac{\hat{odds}_{x+1}}{\hat{odds}_x} = \exp(b_1)$

$\Rightarrow \hat{odds}_{x+1} = \exp(b_1) \cdot \hat{odds}_x$

$\Rightarrow$ **The estimated odds are multiplied by $\exp(b_1)$ for any unit increase in x.**

3.6.2 Example interpreting odds ratio for continuous explanatory variable

Example *Political party*
Here the explanatory variable is a continuous variable (`pro_capital_punishment`)

To obtain the odds ratio, we need to know $\exp(b_1)$

```
glm.log1$coefficients
```

```
##            (Intercept) pro_capital_punishment
##             -1.4477794              0.1751987
```

```
exp(glm.log1$coefficients)
```

```
##            (Intercept) pro_capital_punishment
##              0.2350917              1.1914830
```

Thus $b_1 = 0.175$ and $\exp(b_1) = 1.191$.

The estimated odds are multiplied by 1.20 **for any unit increase in** *pro_capital_punishment*.

Interpretation: The odds of voting Republican is 1.20 times larger for each additional point on the *pro_capital_punishment* score.

Remark:

1. Since $\exp(b_1) = 1.191 > 1$, it indicates that as the predictor increases, the odds of the outcome occurring increase.
2. Consider `subject 1` who has `pro_capital_punishment = 3` and consider `subject 4` who has `pro_capital_punishment = 4` (and hence a one unit increase of the explanatory variable).

```r
head(combine, 5)
```

```
##   pro_capital_punishment Republican Fitted value
## 1                      3          0    0.2845133
## 2                      2          0    0.2502308
## 3                      1          0    0.2188158
## 4                      4          0    0.3214787
## 5                      4          0    0.3214787
```

$Odds\ for\ Republican = \frac{P(Republican=1)}{P(Republican=0)}$

$Odds\ subject\ 4 = 1.20 \cdot Odds\ subject\ 1$
The odds to vote Republican is 1.20 times higher for `subject 4` compared to `subject 1`.

	$P(Republican = 1)$	$P(Republican = 0)$	$Odds$	$Odds\ ratio$
Subject 1	0.28451	0.71549	0.39764	1.1915
Subject 4	0.32148	0.67852	0.47379	

3.7 Simple logistic regression model with categorical explanatory variable

Predictor variables can be categorical. When you want to use these in logistic regression models, you have to be aware of the way R is coding the categories in order to correctly interpret the results.

3.7.1 Use of binary predictor variables

Binary predictor variables should be coded as 0 or 1.

Example *Political party*
The binary predictor *gender* is coded as 0 (female) and 1 (male).

Gender	Coding (data set)
Female	0
Male	1

We use a logistic regression model for $P(Republican = 1)$ with *gender* as only explanatory variable.

```r
# Logistic regression with binary explanatory variable
glm.log2 <- glm(Republican ~ gender, family = binomial(link = "logit"), data = PP)
summary(glm.log2)
```

```
##
## Call:
## glm(formula = Republican ~ gender, family = binomial(link = "logit"),
##     data = PP)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2557  -0.5749  -0.5749   1.1010   1.9400
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.1823     0.1748   1.043    0.297
## gender       -1.8989     0.2861  -6.638 3.19e-11 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 361.18  on 282  degrees of freedom
## Residual deviance: 310.76  on 281  degrees of freedom
## AIC: 314.76
##
## Number of Fisher Scoring iterations: 3
```

With odds ratio

```r
# To obtain the odds ratio
exp(glm.log2$coefficients)
```

```
## (Intercept)      gender
##   1.2000000   0.1497396
```

Interpreting the odds ratio

- The odds ratio to vote Republican for males to females is 0.15 (which is $\exp(-1.9)$).
- The odds to vote Republican for males is 0.15 times the odds to vote Republican for females.
- The odds to vote Republican for females is $\frac{1}{0.15} = 6$ times the odds to vote Republican for males.

Remark:
The logistic regression model is estimated by
$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = 0.182 - 1.9 \cdot gender$ with $p = P(Republican = 1)$.

- $\log(\hat{odds})$ for voting Republican for females ($gender = 0$):
  $\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = 0.182$
- $\log(\hat{odds})$ for voting Republican for males ($gender = 1$):
  $\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = 0.182 - 1.9 = -1.718$

$\log(\hat{odds}_{female})$	0.182
$\hat{odds}_{female}$	1.20
$\log(\hat{odds}_{male})$	-1.718
$\hat{odds}_{male}$	0.18
Odds Ratio (male to female)	0.15
Odds Ratio (female to male)	6.67

We can ask for the estimated probabilities

```r
combine2 <- data.frame(cbind(PP$gender, PP$Republican, fitted(glm.log2)))
colnames(combine2) <- c("gender", "Republican", "Fitted value")
combine2[c(1,18:22),]
```

```
##    gender Republican Fitted value
## 1       0          0    0.5454545
## 18      0          0    0.5454545
## 19      0          0    0.5454545
## 20      0          0    0.5454545
## 21      1          0    0.1523179
## 22      1          0    0.1523179
```

3.7.2 Use of categorical predictor variable (not binary)

Example *Titanic*

For this example, import the data set *titanic.xlsx* as `titanic`.

```
names(titanic)
```

```
## [1] "Class"     "Age"       "Sex"       "survived"  "Class_New"
```

We want to investigate whether the variable *Class_New* can be used as predictor variable for surviving the titanic. The *Class_New* variable is a categorical predictor with 4 levels as indicated below:

Name	Description
Class_New	1: 1^{st} class 2: 2^{nd} class 3: 3^{rd} class 4: crew
survived	0: no 1: yes
Sex	0: female 1: male
Age (group)	0: child 1: adult

Since *Class_New* is numeric, R assumes by default that it is continuous. Therefore, we use the function `as.factor()`

```
titanic$class.f <- as.factor(titanic$Class_New)
glm.log1 <- glm(survived ~ class.f, family = binomial(link = logit), data = titanic)
summary(glm.log1)
```

```
##
## Call:
## glm(formula = survived ~ class.f, family = binomial(link = logit),
##     data = titanic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3999  -0.7623  -0.7401   0.9702   1.6906
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.5092     0.1146   4.445 8.79e-06 ***
## class.f2     -0.8565     0.1661  -5.157 2.51e-07 ***
## class.f3     -1.5965     0.1436 -11.114  < 2e-16 ***
## class.f4     -1.6643     0.1390 -11.972  < 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: 2769.5  on 2200  degrees of freedom
## Residual deviance: 2588.6  on 2197  degrees of freedom
## AIC: 2596.6
##
## Number of Fisher Scoring iterations: 4
```

Then 3 new Dummy variables are created. By default, the first category is the reference category. Here, we want to compare the crew ($Class\_New = 4$). Hence, we take this category as reference category.

```r
# We want to change the reference group.
# We want Class_New = 4 to be the reference category.
titanic$Class_Ref <- relevel(titanic$class.f, ref = "4")
glm.log2 <- glm(survived ~ Class_Ref, family = binomial(link = logit), data = titanic)
summary(glm.log2)
```

```
##
## Call:
## glm(formula = survived ~ Class_Ref, family = binomial(link = logit),
##     data = titanic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3999  -0.7623  -0.7401   0.9702   1.6906
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.15516    0.07876 -14.667  < 2e-16 ***
## Class_Ref1   1.66434    0.13902  11.972  < 2e-16 ***
## Class_Ref2   0.80785    0.14375   5.620 1.91e-08 ***
## Class_Ref3   0.06785    0.11711   0.579    0.562
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2769.5  on 2200  degrees of freedom
## Residual deviance: 2588.6  on 2197  degrees of freedom
## AIC: 2596.6
##
## Number of Fisher Scoring iterations: 4
```

```r
# To obtain the odds ratio
exp(glm.log2$coefficients)
```

```
## (Intercept)  Class_Ref1  Class_Ref2  Class_Ref3
##   0.3150074   5.2822069   2.2430799   1.0702008
```

3.7.2.1 How to interpret the odds ratio?

- Odds ratio of 1^{st} *class* to crew = 5
  The odds to survive the titanic is 5 times larger for passengers from first class than for the crew.
- Odds ratio of 2^{nd} *class* to crew = 2
  The odds to survive the titanic is 2 times larger for passengers from second class than for the crew.
- Odds ratio of 3^{rd} *class* to crew = 1 and is not significant.

3.7.2.2 The model is estimated as

Let $p = P(survived = 1)$, then
$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -1.155 + 1.664 \cdot Ind_{Class1} + 0.808 \cdot Ind_{Class2} + 0.068 \cdot Ind_{Class3}$

- For passengers from *Class 1*:
  $\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -1.155 + 1.664 = 0.509$
- For passengers from *Class 2*:
  $\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -1.155 + 0.808 = -0.347$

- For passengers from *Class 3*: Not significant different than for the crew
  $\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -1.155 + 0.068 = -1.087$

Class	$\log(odds)$	$odds$	$prob$	OR
Class 1	0.509	1.664	0.625	5.28
Class 2	-0.347	0.707	0.414	2.24
Class 3	-1.087	0.337	0.252	1.07
Crew	-1.155	0.315	0.240	

3.8 Goodness of fit

There exists several measures to investigate the goodness-of-fit of your model.

- Chi-square goodness of fit test (to test whether the logistic response function is appropriate - see Hosmer and Lemeshow)
- Wald test of significant coefficients
- Deviance: $-2\cdot$ Log likelihood
- Pseudo R^2
- ROC curve (predictive power of the logistic model)

Example *Political party*
The logistic regression model (with $p = P(Republican = 1)$)
$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \cdot pro\_capital\_punishment$
is estimated by
$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -1.448 + 0.175 \cdot pro\_capital\_punishment$

What is the fit of this logistic model?

```
glm.log1 <- glm(Republican ~ pro_capital_punishment, family = binomial(link = logit), data = PP)
summary(glm.log1)
```

```
##
## Call:
## glm(formula = Republican ~ pro_capital_punishment, family = binomial(link = logit),
##     data = PP)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3090  -0.9461  -0.8183   1.3498   1.6646
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -1.44778    0.38716   -3.74 0.000184 ***
## pro_capital_punishment  0.17520    0.08263    2.12 0.033988 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 361.18  on 282  degrees of freedom
## Residual deviance: 356.62  on 281  degrees of freedom
## AIC: 360.62
##
## Number of Fisher Scoring iterations: 4
```

3.8.1 Hosmer-Lemeshow goodness of fit test

The Hosmer-Lemeshow goodness of fit test assess whether the predicted probabilities match the observed probabilities.

H_0 : **The logistic regression model fits the data.**
versus
H_1 : **The logistic regression model does not provide a good fit.** .

(Hence, here we hope to have a $p - value$ larger than the chosen significance level.)

Step 1: Based on the estimated logistic regression model, calculate the predicted probabilities of success for all observations.

Step 2: Order the data by these predicted probabilities (from small to large).

Step 3: Split the data into (approximately) 10 groups as follows. The first group consists of those observations with the lowest 10% predicted probabilities. The second group consists of the observations with the next 10% lowest predicted probabilities etc.

Reasoning of the Hosmer-Lemeshow test:
Suppose now (artificially) that our total sample size is 100 (and hence we have 10 groups of 10 observations each).

- Suppose now that all observations in the 1^{st} group have predicted probability 0.1.
- Then, if the H_0 is true, we would expect 1 observation that has $Y = 1$.
- If indeed H_0 is true, the observed proportion of observations with $Y = 1$ in that group will be around 0.1.
- In case we would have observed 8 observations in that 1^{st} group with $Y = 1$ then this would suggest that the model was not fitting the data well.

Step 4:

- Compute in each group the expected number of observations with $Y = 1$ and the observed number of observations with $Y = 1$.
- Compute in each group the expected number of observations with $Y = 0$ and the observed number of observations with $Y = 0$.

Remark:
How to compute the expected number of observations with $Y = 1$?
In practice, each observation in a group will have a different predicted probability.

- In every group we compute the average of the predicted probabilities for that group $(Y = 1) = \hat{\pi}_i$

- In every group, we can compute the expected number of observations with $(Y = 1) = n_i\hat{\pi}_i$, with n_i the number of observations in group i.

Step 5: We compute the Pearson goodness of fit statistic and the corresponding p-value.
Test statistic:
$\sum_{i=1}^{10} \frac{(O_{1i}-E_{1i})^2}{E_{1i}} + \frac{(O_{0i}-E_{0i})^2}{E_{0i}} \sim \chi^2_8$
with:

- O_{1i} the observed number of $(Y = 1)$ in the i^{th} group.
- E_{1i} the observed number of $(Y = 1)$ in the i^{th} group.
- O_{0i} the observed number of $(Y = 0)$ in the i^{th} group.
- E_{0i} the observed number of $(Y = 0)$ in the i^{th} group.

In R

The function `hoslem.test` (from package `ResourceSelection`) executes the Hosmer-Lemeshow goodness of fit test.

```r
install.packages("ResourceSelection")
library(ResourceSelection)
```

```r
Republican <- PP$Republican
hoslem <- hoslem.test(Republican, fitted(glm.log1))
hoslem
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  Republican, fitted(glm.log1)
## X-squared = 14.594, df = 8, p-value = 0.06754
```

```r
combine <- cbind (hoslem$observed, hoslem$expected)
combine
```

```
##                 y0 y1    yhat0     yhat1
## [0.219,0.25]  23  9 24.11828  7.881725
## (0.25,0.285]  36 22 41.49823 16.501769
## (0.285,0.321] 48 20 46.13945 21.860553
## (0.321,0.361] 57 15 46.02061 25.979391
## (0.361,0.402] 17 14 18.53389 12.466112
## (0.402,0.575]  7 15 11.68955 10.310450
```

Remark:
This test is not powerful when you have a small number of observations. You can only trust the $p-value$ if the underlying assumption for a Pearson chi-square statistic is satisfied. This assumes that the expected number of observations in each cell is at least 5 (at least in 20% of the cells).

3.8.2 Wald test to test significance of regression coefficients

Wald test is used to test the statistical significance of each covariate in the model.

Statement of hypotheses:
$H_0 : \beta_j = 0$
versus
$H_1 : \beta_j \neq 0$.

The test statistic of the Wald test is
$W = \frac{Estimate}{Standard\ Error}$
Under the null hypothesis, $W \sim N(0, 1)$.

Example *Political party*

Statement of hypotheses:
$H_0 : \beta_{pro\_capital\_punishment} = 0$
versus
$H_1 : \beta_{pro\_capital\_punishment} \neq 0$.

```r
summary(glm(formula = Republican ~ pro_capital_punishment, family = binomial(link = logit),
            data = PP))$coefficients
```

```
##                          Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)            -1.4477794 0.38715651 -3.739520 0.0001843721
## pro_capital_punishment  0.1751987 0.08263244  2.120218 0.0339877032
```

$p-value = 0.034 < 0.05$, hence $pro\_capital\_punishment$ is a significant variable in this logistic model.

3.8.3 Deviance

$Deviance = -2 \cdot$ (Log-likelihood of fitted model)

Deviance is a statistic that compares the *log-likelihood of the fitted model* to the *log-likelihood of a saturated model*.

A **saturated model** is a model with n parameters that fits the n observations.
$\rightarrow$ n parameters for n observations
$\rightarrow$ perfect fit! (residuals will all be zero)
$\rightarrow$ **Log-likelihood for a saturated model** $= 0$**.**

Compare this log-likelihood value for the saturated model ($= 0$) with the log-likelihood value for the fitted model.

A **fitted model** is a logit model with less parameters than in the saturated model.
$\rightarrow$ # parameters in fitted model $<$ # parameters in saturated model
$\rightarrow$ log-likelihood fitted model $<$ log-likelihood saturated model ($= 0$)

We now look at the difference between both (=deviance)
Deviance
$= 2 \cdot$ (Log-likelihood of saturated model) $- 2 \cdot$ (Log-likelihood of fitted model)
$= 0 - 2 \cdot$ (Log-likelihood of fitted model)
$\rightarrow$ This difference is always positive

The smaller the *deviance*($= -2 \cdot$ (Log-likelihood of fitted model)), the closer the fitted model is to the saturated model.
$\rightarrow$ This statistic **can be used as a goodness of fit criterion!**

The larger the *deviance*($= -2 \cdot$ (Log-likelihood of fitted model)), the poorer the fit is between the fitted model and the saturated model.

Example *Political party*

```
summary(glm(formula = Republican ~ pro_capital_punishment, family = binomial(link = logit),
            data = PP))
```

```
##
## Call:
## glm(formula = Republican ~ pro_capital_punishment, family = binomial(link = logit),
##     data = PP)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3090  -0.9461  -0.8183   1.3498   1.6646
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -1.44778    0.38716   -3.74 0.000184 ***
## pro_capital_punishment  0.17520    0.08263    2.12 0.033988 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 361.18  on 282  degrees of freedom
## Residual deviance: 356.62  on 281  degrees of freedom
## AIC: 360.62
##
```

```
## Number of Fisher Scoring iterations: 4
```

$-2 \cdot (\text{Log-likelihood of fitted model}) = 356.62$

In R: the deviance ($-2 \cdot (\text{Log-likelihood of fitted model})$) is also given and can be seen as a generalization of the residual (or error) sum of squares (regression analysis). It is often used as a measure to compare several models, each a subset of the other, and to test whether the model with more terms is significantly better than the model with fewer terms.

3.8.4 Pseudo R^2

- **In regression analysis**, the R^2 represents that proportion of variance which is explained by the regression model
  $R^2 = \frac{modelSS}{TotalSS} = \frac{TotalSS - ErrorSS}{TotalSS}$
- **In logistic regression**, it is not possible to compute an R^2 but we can define something similar. It expresses the proportional reduction in the log-likelihood measure. It **measures how the badness of fit improves** as a result of including explanatory variables.

$Pseudo\ R^2 = 1 - \frac{-2 \cdot (\text{Log-likelihood of fitted model})}{-2 \cdot (\text{Log-likelihood of null model})}$

The *null model* is the model with only the intercept.

In R
Computing pseudo R^2 in R with function `pR2()` from the package `pscl`

```r
library(pscl)
pR2(glm.log1)
```

```
## fitting null model for pseudo-r2

##           llh       llhNull            G2      McFadden          r2ML
## -178.30939464 -180.59207832    4.56536736    0.01264000    0.01600262
##          r2CU
##    0.02219739
```

We only have a small value of 0.012 which means that we can only explain a small part of the deviance by the variable *pro_capital_punishment*.

3.9 Classification of observations

Example *Political party*
Dependent variable: *Republican*
Predictor variable: *pro_capital_punishment*

```r
glm.log1 <- glm(Republican ~ pro_capital_punishment, family = binomial(link = logit), data = PP)
combine <- data.frame(cbind(PP$pro_capital_punishment, PP$Republican, fitted(glm.log1)))
colnames(combine) <- c("pro_capital_punishment", "Republican", "Fitted value")
head(combine,5)
```

```
##   pro_capital_punishment Republican Fitted value
## 1                      3          0    0.2845133
## 2                      2          0    0.2502308
## 3                      1          0    0.2188158
## 4                      4          0    0.3214787
## 5                      4          0    0.3214787
```

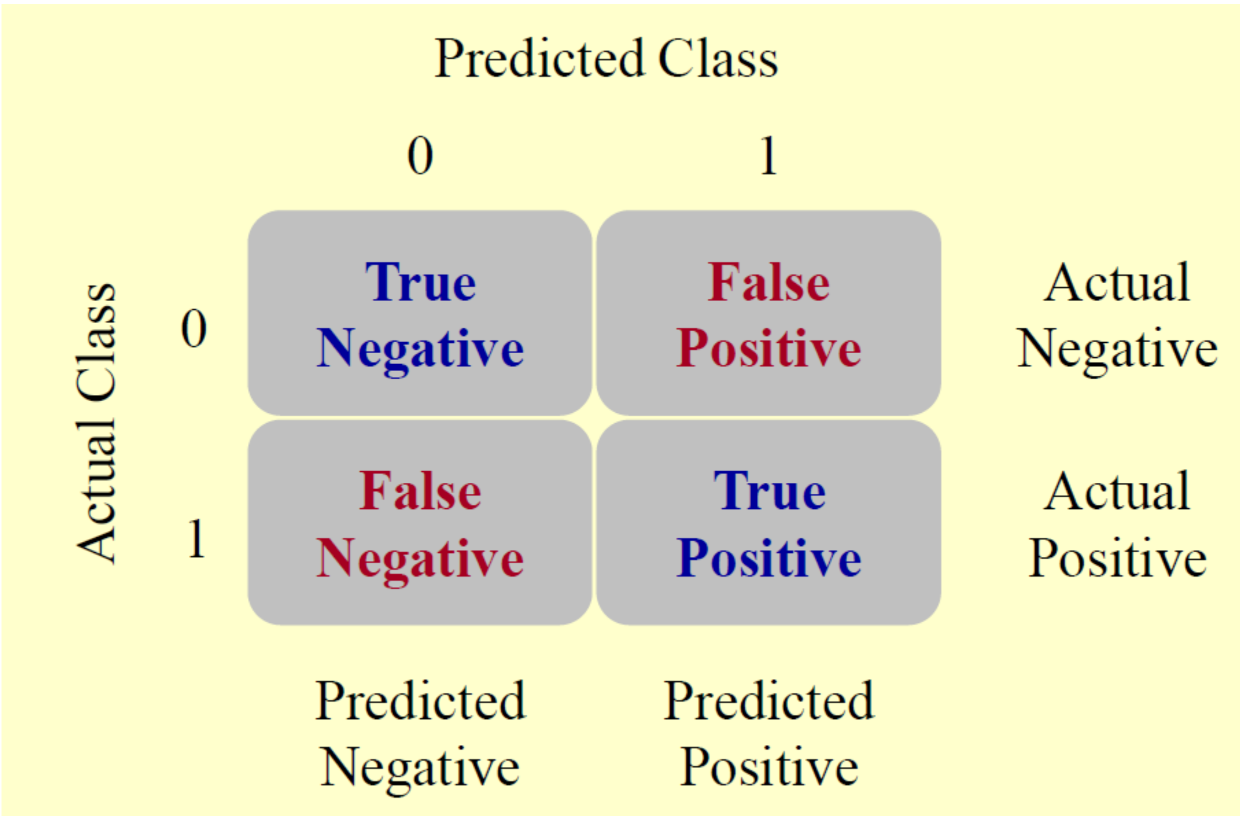
Before observations can be classified, the probabilities needs to be estimated. By using the fitted model, the estimated probability (predicted value) can be computed for each observation. Next, these probabilities can be used to classify observations into two groups.

If predicted probability > 0.5 then observation is classified as voting Republican ($pred\_group = 1$).
If predicted probability < 0.5 then observation is classified as not voting Republican ($pred\_group = 0$).

Classification table:

```
table(Republican, fitted(glm.log1) > 0.5)
```

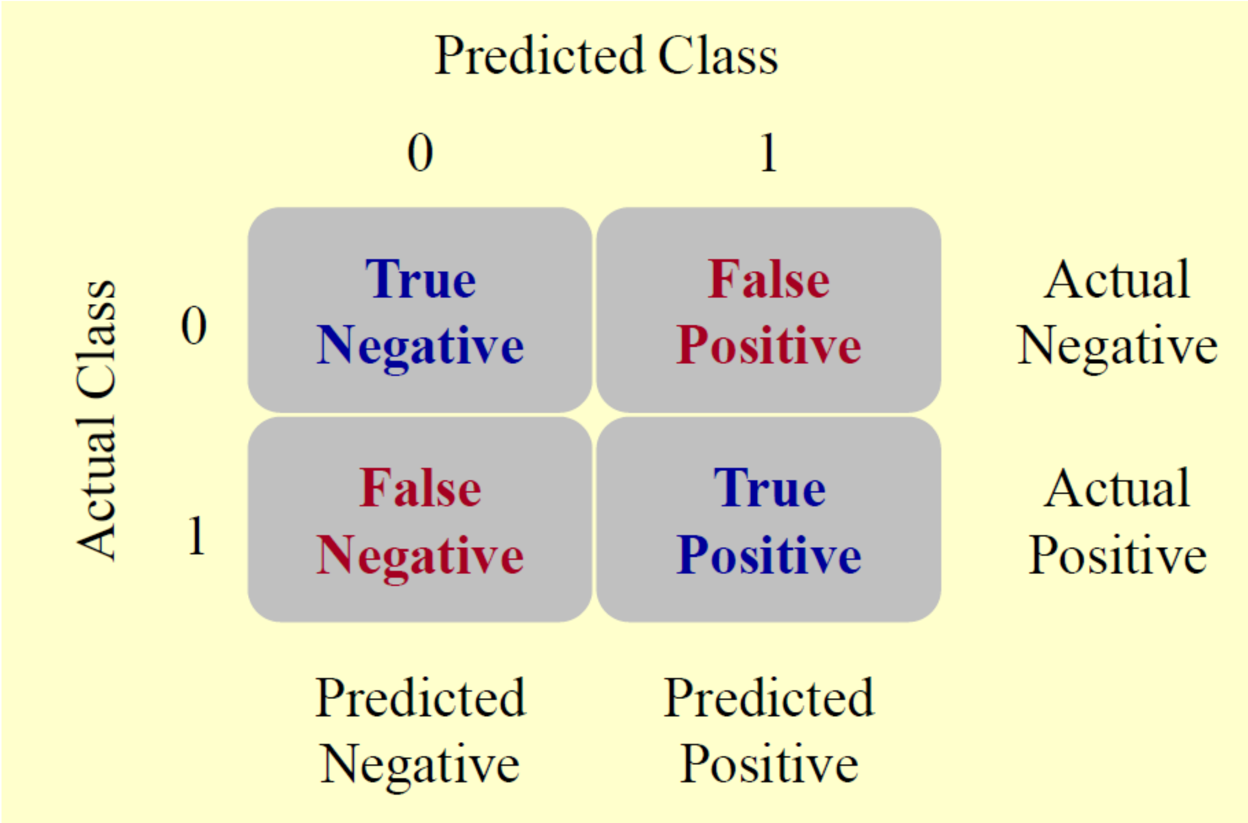
```
##
## Republican FALSE TRUE
##          0   187    1
##          1    93    2
```



3.10 ROC curve

3.10.1 What is a ROC curve?

The Receiving Operating Characteristic (ROC) curves are graphs that are used to evaluate and compare the performance of classification models. The **ROC curve** is a visual measure for the predictive ability of the (logistic) regression model. The area under the ROC curve (which is abbreviated as AUC) indicates the performances of a binary classifier in a single value.



The following terms are important for understanding the ROC curve:

- **False positive**: Non-event (actual class = 0) which is predicted as event (predicted class = 1).
- **False negative**: Event (actual class = 1) which is predicted as non-event (predicted class = 0).
- **Sensitivity**: Proportion of events (actual class = 1) which are predicted as events (predicted class = 1). The sensitivity is also referred to as *true positive rate*.
  $Sensitivity = \frac{True\ positives}{True\ positives + False\ negatives}$
- **Specificity**: Proportion of non-events (actual class = 0) which are predicted as non-events (predicted class = 0).
  $Specificity = \frac{True\ negatives}{True\ negatives + False\ positives}$
- The **false positive rate** is proportion of non-events (actual class = 0) got incorrectly classified by the classifier.
  $false\ positive\ rate = 1 - specificity = \frac{False\ postives}{True\ negatives + False\ positives}$

These values vary according to the chosen cut-off value.

Example *Political party*

1. We have obtained this classification table for a *cut-off value of 0.5*.

Classification table:

```
table(Republican, fitted(glm.log1) > 0.5)
```

```
##
## Republican FALSE TRUE
##          0   187    1
##          1    93    2
```

$Sensitivity = \frac{2}{2+93} = 0.021$
$Specificity = \frac{187}{187+1} = 0.995$

$False\ positive\ rate = \frac{1}{187+2} = 0.005$

2. For a *cut-off value of 0.9*.
   A high-cut off value implies that almost everything is predicted as a non-event. Hence sensitivity will be small and false positive will be small.

Classification table:

```
table(Republican, fitted(glm.log1) > 0.9)
```

```
##
## Republican FALSE
##          0   188
##          1    95
```

$Sensitivity = \frac{0}{95} = 0$
$Specificity = \frac{188}{188} = 1$
$False\ positive\ rate = \frac{0}{188} = 0$

3. For a *cut-off value of 0.1*.
   A low-cut off value implies that almost everything is predicted as a success. Hence sensitivity will be high and false positive will be high.

Classification table:

```
table(Republican, fitted(glm.log1) > 0.1)
```

```
##
## Republican TRUE
##          0  188
##          1   95
```

$Sensitivity = \frac{95}{95} = 1$
$Specificity = \frac{0}{188} = 0$
$False\ positive\ rate = \frac{188}{188} = 1$

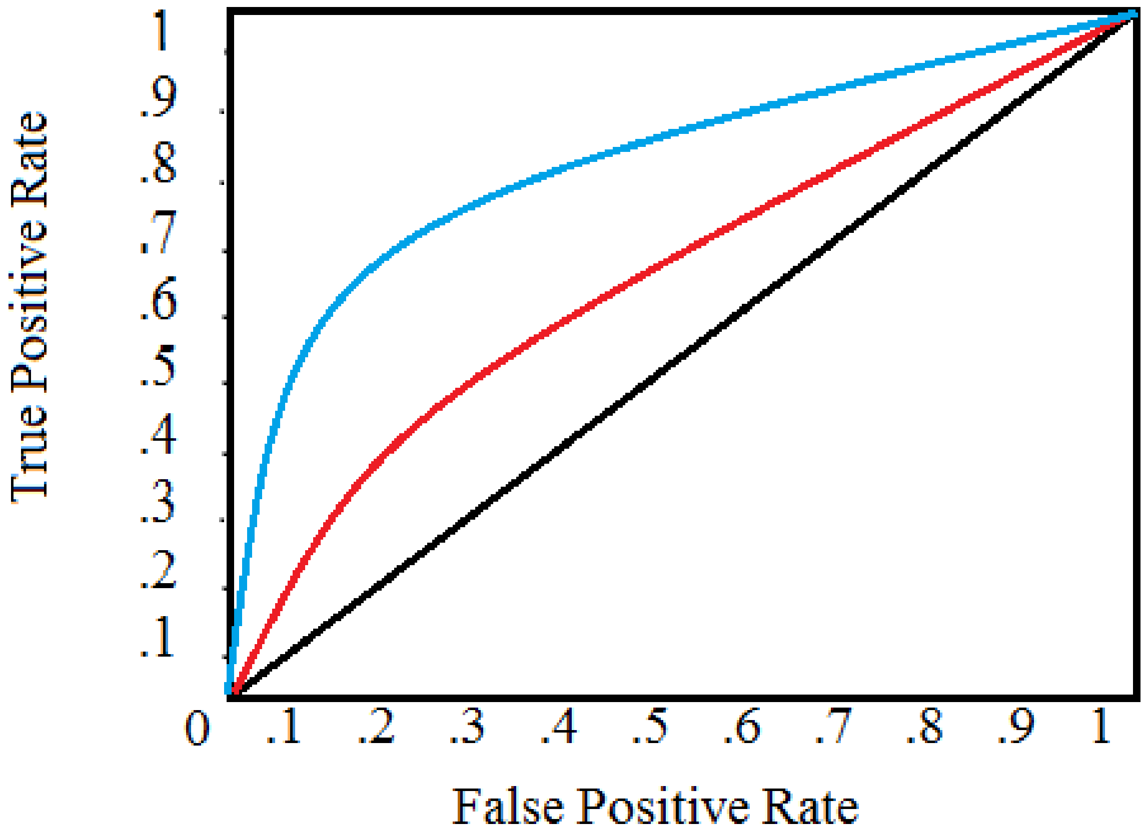
4. The optimal solution would be to have:
   - A small proportion of false positive
   - A large number of sensitivity

Once we have computed sensitivity and specificity pairs for each possible cutoff point, the ROC curve is a plot of sensitivity on the y axis by false positive rate (=1-specificity) on the x axis.

This curve is called the receiver operating characteristic (ROC) curve. The area under the ROC curve ranges from 0.5 and 1.0 where larger values indicate a better fit.

The image below shows ROC curves of a few logistic regression models. [1]

[1] Figure is from https://www.statisticshowto.com/receiver-operating-characteristic-roc-curve/.

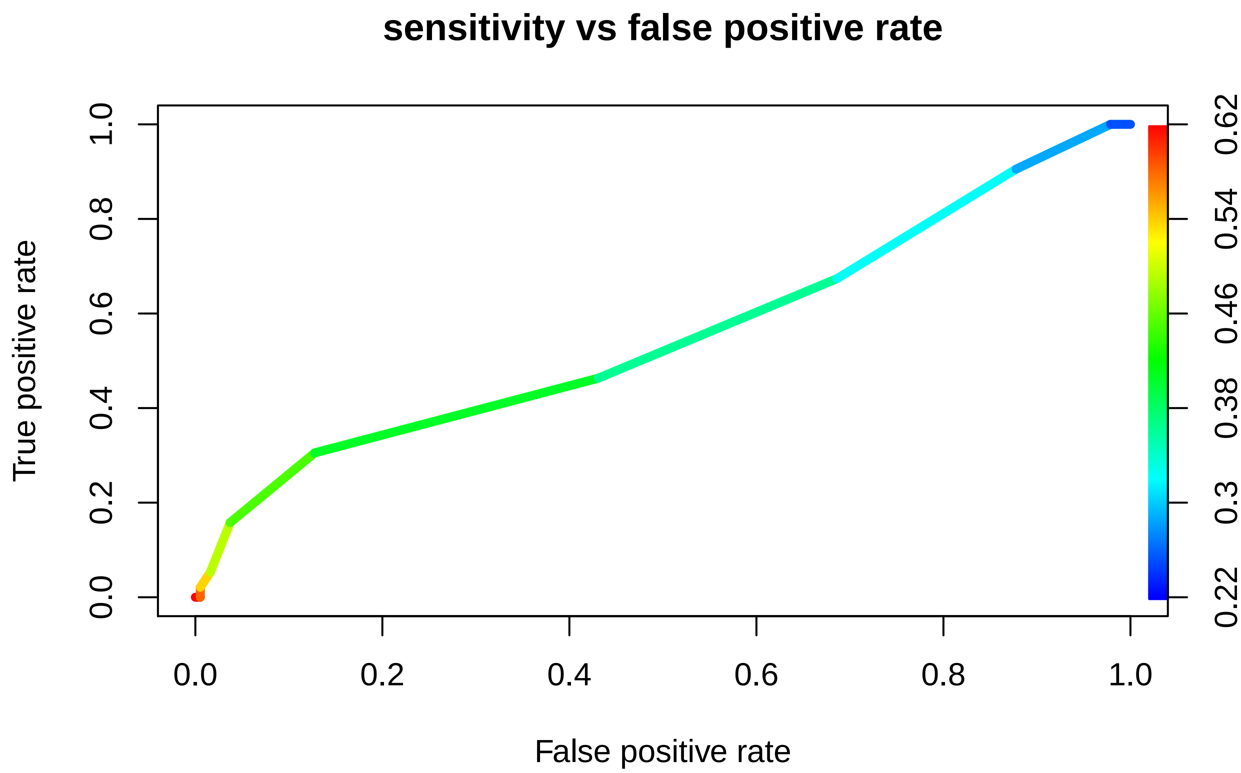


The classifier corresponding to the red curve is less accurate than the classifier corresponding to the blue curve.

3.10.2 How to obtain the ROC curve in R

```
install.packages("ROCR")
library(ROCR)
```

```
glm.log1 <- glm(Republican ~ pro_capital_punishment, family = binomial(link = logit), data = PP)
predict <- fitted(glm.log1)
pred <- prediction(predict, Republican)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
# Colorize argument in following plot function:
# This logical determines whether the curve(s)
# should be colorized according to cutoff
plot(perf, main = "sensitivity vs false positive rate", colorize = TRUE,
     colorkey.relwidth = 0.5, lwd = 4.5)
```



X-axis: False positive rate = 1 - specificity
Y-axis: True positive rate = sensitivity

Area under the ROC curve:

```
perf_auc <- performance(pred, measure = "auc")
perf_auc@y.values
```

```
## [[1]]
## [1] 0.5539194
```

We here have an AUC (area under the ROC curve) of 0.554 which is not good. The model does not have a good discriminating ability.

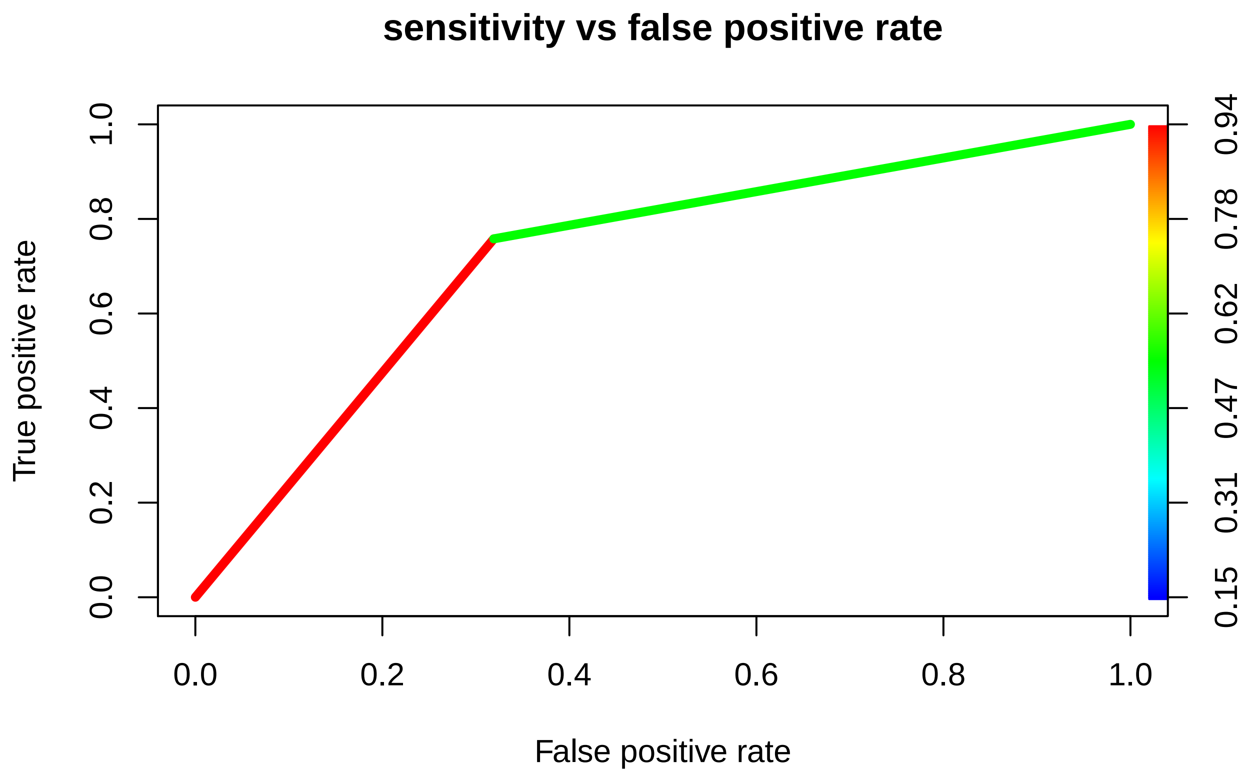
Remark:
Models with a higher predictive power has a higher AUC.

3.10.3 Example

Example *Political party*

Dependent variable: *Republican*
Predictor variable: *gender*
Plot the ROC curve and compute the AUC.

```
glm.log2 <- glm(Republican ~ gender, family = binomial(link = "logit"), data = PP)
pred2 <- prediction(fitted(glm.log2), Republican)
perf2 <- performance(pred2, measure = "tpr", x.measure = "fpr")
plot(perf2, main = "sensitivity vs false positive rate", colorize = TRUE,
     colorkey.relwidth = 0.5, lwd = 4.5)
```

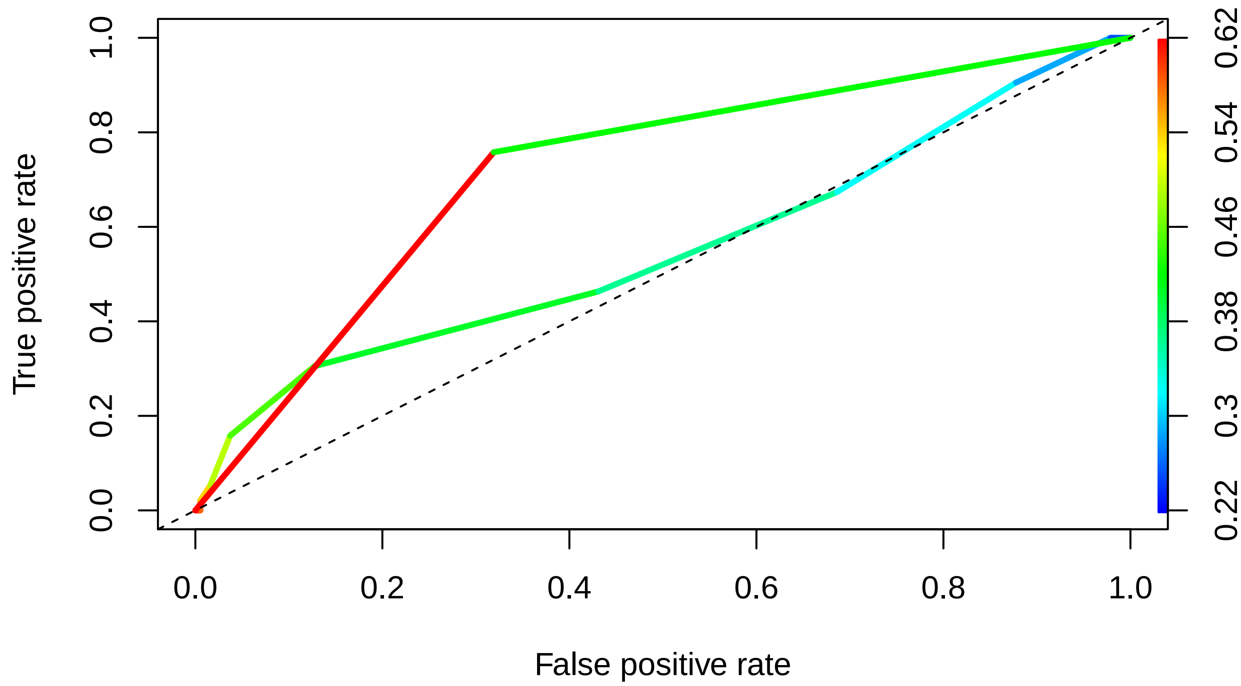



```r
performance(pred2, measure = "auc")@y.values
```

```
## [[1]]
## [1] 0.7193729
```

Remark:
In case you want to show two ROC curves on the same plot:

```r
plot(perf, colorize = TRUE, lwd = 3)
plot(perf2, add = TRUE, colorize = TRUE, lwd = 3)
abline(0, 1, lty = 2)
```



4 Multiple logistic regression

4.1 General

In **simple logistic regression**, we have only 1 predictor variable:
$P(Y=1) = E(Y) = \frac{\exp(\beta_0+\beta_1 x)}{1+\exp(\beta_0+\beta_1 x)}$

In case we have **more predictor variables** (e.g., p), the model becomes
$P(Y=1) = E(Y) = \frac{\exp(\beta_0+\beta_1 x_1+\beta_2 x_2+...+\beta_p x_p)}{1+\exp(\beta_0+\beta_1 x_1+\beta_2 x_2+...+\beta_p x_p)}$.

Or the model can be written as
$P(Y=1) = \frac{1}{1+\exp(-\beta_0-\beta_1 x_1-\beta_2 x_2-...-\beta_p x_p)}$.

By using the *logit transformation*
$p' = \log_e\left(\frac{p}{1-p}\right)$
we obtain the *logit response function*:
$p' = \log_e\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$.

Properties:

- monotonic and sigmoid in shape
- Almost linear when p is between 0.2 and 0.8
- Predictor variables may be interaction effects, curvature, quantitative qualitative.
- A logistic regression model with only qualitative variables is called a *log-linear* model
- Maximum likelihood estimation is used to find estimates for the parameters.

4.2 Example

Example *Political party*

We now perform a logistic regression analysis with 4 explanatory variables of which 3 scale variables ($pro\_capital\_punishment, pro\_welfare\_reform$ and $pro\_fed\_support\_ed$) and 1 indicator variable ($gender$).

4.2.1 Hierarchical step by step (manually)

Model A: model with 4 explanatory variables

```
glm.log.A <- glm(Republican ~ pro_capital_punishment + pro_welfare_reform +
                   pro_fed_support_ed + gender, family = binomial(link = logit),
                 data = PP)
summary(glm.log.A)
```

```
##
## Call:
## glm(formula = Republican ~ pro_capital_punishment + pro_welfare_reform +
##     pro_fed_support_ed + gender, family = binomial(link = logit),
##     data = PP)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0351  -0.7323  -0.3916   0.8786   2.2831
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -2.84174    0.96470  -2.946  0.00322 **
## pro_capital_punishment  0.67520    0.13240   5.100 3.40e-07 ***
## pro_welfare_reform      0.06017    0.10056   0.598  0.54963
## pro_fed_support_ed      0.04408    0.11274   0.391  0.69580
## gender                 -3.07436    0.41479  -7.412 1.25e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 361.18  on 282  degrees of freedom
## Residual deviance: 273.97  on 278  degrees of freedom
## AIC: 283.97
##
## Number of Fisher Scoring iterations: 5
```

Model B: model with 3 explanatory variables

Since Federal Support Education is not significant, we drop this variable from the model and refit the model.

```
glm.log.B <- glm(Republican ~ pro_capital_punishment + pro_welfare_reform + gender,
                 family = binomial(link = logit), data = PP)
summary(glm.log.B)
```

```
##
## Call:
## glm(formula = Republican ~ pro_capital_punishment + pro_welfare_reform +
##     gender, family = binomial(link = logit), data = PP)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -3.0417  -0.7337  -0.3903   0.8987   2.2858
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -2.60435    0.74361  -3.502 0.000461 ***
## pro_capital_punishment  0.68069    0.13209   5.153 2.56e-07 ***
## pro_welfare_reform      0.05909    0.10043   0.588 0.556316
## gender                  -3.06831    0.41452  -7.402 1.34e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 361.18  on 282  degrees of freedom
## Residual deviance: 274.12  on 279  degrees of freedom
## AIC: 282.12
##
## Number of Fisher Scoring iterations: 5
```

Model C: model with 2 explanatory variables
Since $pro\_welfare\_reform$ is not significant, we drop this variable from the model and refit the model.

```
glm.log.C <- glm(Republican ~ pro_capital_punishment + gender,
                 family = binomial(link = logit), data = PP)
summary(glm.log.C)
```

```
##
## Call:
## glm(formula = Republican ~ pro_capital_punishment + gender, family = binomial(link = logit),
##     data = PP)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0501  -0.7243  -0.3807   0.9068   2.3068
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -2.2777     0.4873  -4.675 2.95e-06 ***
## pro_capital_punishment   0.6920     0.1308   5.290 1.22e-07 ***
## gender                   -3.0782     0.4143  -7.429 1.09e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 361.18  on 282  degrees of freedom
## Residual deviance: 274.47  on 280  degrees of freedom
## AIC: 280.47
##
## Number of Fisher Scoring iterations: 5
```

In this model, all variables are significant.

4.2.2 Comparison of several models

Suppose you want to compare the following models

- Model 0: Intercept only
- Model 1: $gender$
- Model 2: $gender + pro\_capital\_punishment$
- Model 3: $gender + pro\_capital\_punishment + pro\_welfare\_reform$
- Model 4: $gender + pro\_capital\_punishment + pro\_welfare\_reform + pro\_fed\_support\_ed$

Source	Deviance ($= -2 \cdot$ log-likelihood)	pseudo R^2
Model 0: Intercept only	361.184	
Model 1: $gender$	310.764	0.16
Model 2: $gender + pro\_capital\_punishment$	274.472	0.26
Model 3: $gender + pro\_capital\_punishment + pro\_welfare\_reform$	274.124	0.24
Model 4: $gender + pro\_capital\_punishment + pro\_welfare\_reform + pro\_fed\_support\_ed$	273.971	0.24

Comparing models by comparing the deviances

- For each fitted model, the deviance is calculated, which is $-2 \cdot$ Log-Likelihood.
- Difference between the deviance for two fitted models can be used to compare two nested models. This concept is explained in the next topic.

4.3 Partial deviance

4.3.1 General

1. **Full logistic model**: model with response function (and $p-1$ predictor variables). Where
   $E(Y) = \frac{\exp(\mathbf{x}'\beta_{\mathbf{F}})}{1+\exp(\mathbf{x}'\beta_{\mathbf{F}})}$
   with $\mathbf{x}'\beta_{\mathbf{F}} = \beta_0 + \beta_1 x_1 + ... + \beta_{p-1} x_{p-1}$
   Deviance for the *full model*: $DEV(x_1, ..., x_{p-1})$

2. **reduced logistic model: model with only $q-1$ predictor variables where $q < p$. Where
   $E(Y) = \frac{\exp(\mathbf{x}'\beta_{\mathbf{R}})}{1+\exp(\mathbf{x}'\beta_{\mathbf{R}})}$
   with $\mathbf{x}'\beta_{\mathbf{R}} = \beta_0 + \beta_1 x_1 + ... + \beta_{q-1} x_{q-1}$
   Deviance for the *reduced model*: $DEV(x_1, ..., x_{q-1})$

3. We want to check whether we can drop a set of predictor variables by formulating the null hypothesis:
   $H_0 : \beta_q = \beta_{q+1} = ... = \beta_{p-1} = 0 \qquad \text{with } (q < p).$
   versus
   H_1 : **not all β_k in H_0 are equal to zero.**

If $DEV_{reduced}$ is not much larger than DEV_{full}
$\rightarrow$ reduced model provides almost as close a fit as the full model $\rightarrow$ Do not reject H_0

If $DEV_{reduced}$ is much larger than DEV_{full}
$\rightarrow$ reduced model provides much worse fit compared to the full model $\rightarrow$ Reject H_0

Partial deviance $= DEV(x_1, ..., x_{q-1}) - DEV(x_1, ..., x_{p-1})$

Properties:

- If H_0 holds and n is large, then $partial\ deviance \sim \chi^2_{p-q}$
- Decision rule:
  We compute the partial deviance and the corresponding p-value
  - If $p - value < 0.05$, then reject H_0
  - If $p - value > 0.05$, then do not reject H_0

4.3.2 Example

Example *Political party*

Source	Deviance ($= -2 \cdot$ log-likelihood)	pseudo R^2
Model 0: Intercept only	361.184	
Model 1: $gender$	310.764	0.16
Model 2: $gender +$ $pro\_capital\_punishment$	274.472	0.26
Model 3: $gender +$ $pro\_capital\_punishment +$ $pro\_welfare\_reform$	274.124	0.24
Model 4: $gender +$ $pro\_capital\_punishment$ $+pro\_welfare\_reform +$ $pro\_fed\_support\_ed$	273.971	0.24

```
glm.log.M1 <- glm(Republican ~ gender, family = binomial(link = logit), data = PP)
glm.log.M2 <- glm(Republican ~ gender + pro_capital_punishment,
                  family = binomial(link = logit), data = PP)
glm.log.M3 <- glm(Republican ~ gender + pro_capital_punishment + pro_welfare_reform,
                  family = binomial(link = logit), data = PP)
glm.log.M4 <- glm(Republican ~ gender + pro_capital_punishment + pro_welfare_reform +
                    pro_fed_support_ed, family = binomial(link = logit), data = PP)
```

a) Compare model 2 to model 1
In model 1, we have *gender* as explanatory variable. In model 2, we have *gender* and *pro_capital_punishment* as explanatory variables. We are interested in the improvements of model 2 over model 1.

$H_0 : \beta_{pro\_capital\_punishment} = 0$
versus
$H_1 : \beta_{pro\_capital\_punishment} \neq 0$

Difference in deviance:
$Partial\ deviance = 310.8 - 274.5 = 36.3.$
This is the change in the deviance resulting from adding the variable *pro_capital_punishment* to the model.

Compare several models:

```
anova(glm.log.M1, glm.log.M2, test = "Chisq") # Note the argument 'test = "Chisq"'!
```

```
## Analysis of Deviance Table
##
## Model 1: Republican ~ gender
## Model 2: Republican ~ gender + pro_capital_punishment
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       281     310.76
## 2       280     274.47  1   36.292 1.698e-09 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We have a $p - value$ of 0.00001 which is smaller than 0.05. Hence, the variable $pro\_capital\_punishment$ should be added to the model because it improves the model.

b) Compare model 3 to model 2

$H_0 : \beta_{pro\_welfare\_reform} = 0$
versus
$H_1 : \beta_{pro\_welfare\_reform} \neq 0$

Difference in deviance \$ = 0.348\$

```r
anova(glm.log.M2, glm.log.M3, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Republican ~ gender + pro_capital_punishment
## Model 2: Republican ~ gender + pro_capital_punishment + pro_welfare_reform
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       280     274.47
## 2       279     274.12  1  0.34849    0.555
```

We have a $p-value$ of 0.555. Since $p-value > 0.05$ the variable $pro\_welfare\_reform$ should not be added to the model because it has virtually no effect on the fit (the deviance has hardly changed).

c) Compare model 4 to model 2

$H_0 : \beta_{pro\_fed\_support\_ed} = \beta_{pro\_welfare\_reform} = 0$
versus
$H_1 : \beta_{pro\_fed\_support\_ed} \neq 0$ or $\beta_{pro\_welfare\_reform} \neq 0$ (At least one of these coefficients is not 0).

Difference in deviance $= 274.472 - 273.971 = 0.501$.
Degrees of freedom $= 4 - 2 = 2$

```r
anova(glm.log.M2, glm.log.M4, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Republican ~ gender + pro_capital_punishment
## Model 2: Republican ~ gender + pro_capital_punishment + pro_welfare_reform +
##     pro_fed_support_ed
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       280     274.47
## 2       278     273.97  2  0.50122   0.7783
```

Compute the corresponding $p - value$ as $P(\chi^2_2 \geq 0.501) = 0.778$. The $p - value$ is large indicating that we can drop $pro\_welfare\_reform$ and $pro\_fed\_support\_ed$ from the model.

The optimal model is model 2 with $gender$ and $pro\_capital\_punishment$.

4.4 Interpreting the output

4.4.1 Interpreting the parameter estimates

Example *Political party*

```r
summary(glm.log.M2)$coefficients
```

```
##                          Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)            -2.2777137  0.4872583 -4.674551 2.945979e-06
## gender                 -3.0782127  0.4143275 -7.429419 1.090758e-13
```

```
## pro_capital_punishment  0.6919594  0.1307984  5.290274 1.221335e-07
```

```r
exp(glm.log.M2$coefficients)
```

```
##            (Intercept)                 gender pro_capital_punishment
##             0.10251833             0.04604147             1.99762585
```

The estimated logistic regression function is
$\hat{p} = P(Republican = 1) = \frac{\exp(-2.278 - 3.078 \cdot gender + 0.6920 \cdot pro\_captial\_punishment)}{1 + \exp(-2.278 - 3.078 \cdot gender + 0.6920 \cdot pro\_captial\_punishment)}$

$p' = \log_e\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$

$\hat{p}' = -2.278 - 3.078 \cdot gender + 0.6920 \cdot pro\_captial\_punishment$

- Coefficient for *gender*: $b_{gender} = -3.078$ odds ratio for *gender*: $OR_{gender} = \exp(b_{gender}) = 0.046$

The odds to vote Republican for male ($gender = 1$) is 0.05 times the odds to vote Republican for female ($gender = 0$), when taking *pro_captial_punishment* into account.
OR The odds to vote Republican for female is 20 times the odds to vote Republican for male, when taking *pro_captial_punishment* into account.

- Coefficient for *pro_capital_punishment*: $b_{pcp} = 0.692$ odds ratio for *pro_capital_punishment*: $OR_{pcp} = \exp(b_{pcp}) = 2.00$

Per one unit increase on the score of *pro_capital_punishment*, the odds for voting Republican is increasing 2 times, taking *gender* into account.

4.4.2 Classification table

```r
table(Republican, fitted(glm.log.M2)>0.5)
```

```
##
## Republican FALSE TRUE
##          0   165   23
##          1    50   45
```

4.4.3 Generalized R^2 value

Mc Fadden R^2

```r
library(pscl)
pR2(glm.log.M2)
```

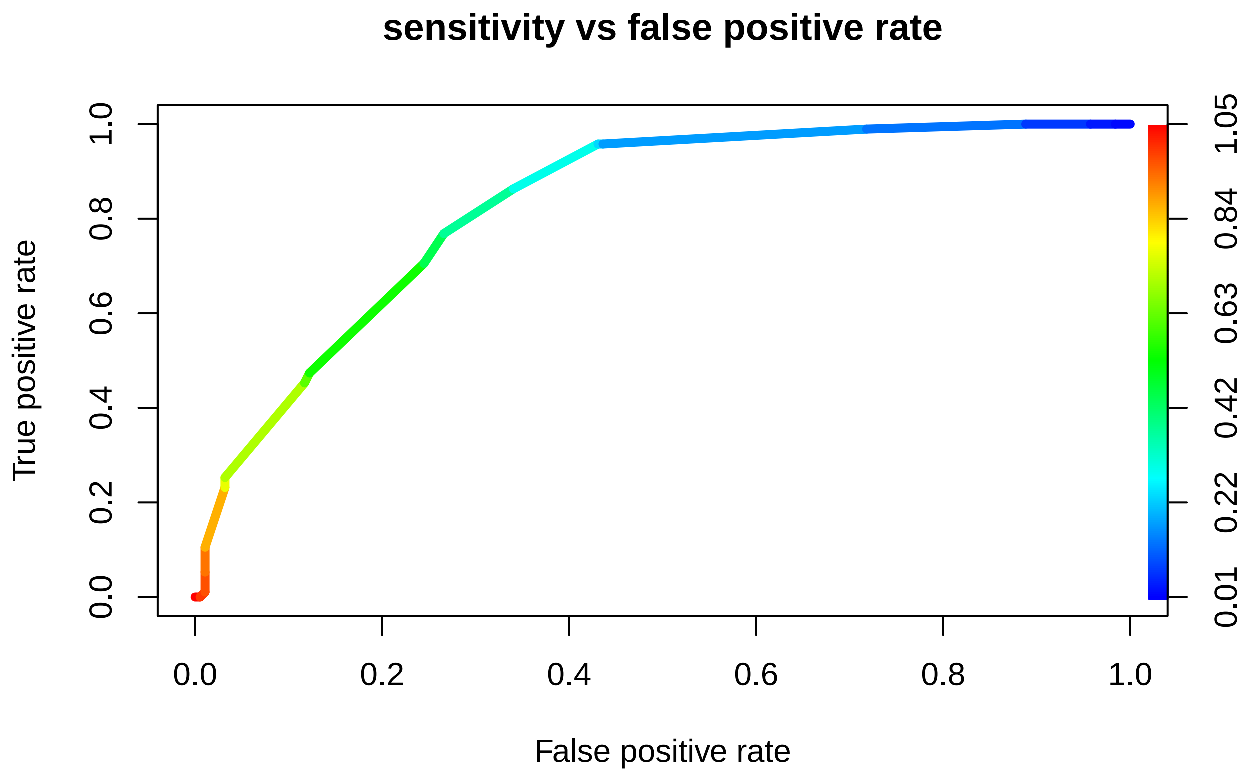
```
## fitting null model for pseudo-r2
```

```
##          llh      llhNull           G2     McFadden         r2ML         r2CU
## -137.2361584 -180.5920783   86.7118398    0.2400765    0.2639095    0.3660715
```

Pseudo R^2 value is 0.24.

4.4.4 Create the ROC curve and area under the curve

```r
pred.M2 <- prediction(fitted(glm.log.M2), Republican)
perf.M2 <- performance(pred.M2, measure = "tpr", x.measure = "fpr")
plot(perf.M2, main = "sensitivity vs false positive rate",
     colorize = TRUE, colorkey.relwidth = 0.5, lwd = 4.5)
```




```r
performance(pred.M2, measure = "auc")@y.values
```

```
## [[1]]
## [1] 0.8275756
```

The AUC is now 0.828

5 References

Meyers, L. S., Gamst, G. & Guarino, A.J. (2017) Applied Multivariate research, Design and Interpretation, 3rd ed., Sage Edge