Generalized Linear Model

Emma Li

Linear Model (LM)

Linear Model makes several key assumptions:

- Linear relationship between X and $E(Y) = \mu = \mathbf{X} * \beta$
- Multivariate normality
- No or little multicollinearity
- No auto-correlation
- Error terms have similar variances

$$\mathbf{E(Y)} = \mu = \mathbf{X} * \beta$$

Generalized Linear Model (GLM)

Generalized Linear Model is the general case of linear regression. It allows Y to have error distribution model other than normal distribution.

Key Assumptions:

- Linear relationship between X and g(E(Y)) = $\mathbf{g(}\ \mu\ \mathbf{) = X *}\ \beta$

Depending on the distribution, we have a link function g()

- No or little multicollinearity
- No auto-correlation
- Error terms have similar variances

$$\mathbf{g(\ E(Y)\ ) = g(}\ \mu\ \mathbf{) = X *}\ \beta$$

Distributions in Exponential Family

Y can follow normal distribution, Bernoulli distribution, binomial distribution, Poisson distribution, negative binomial distribution, gamma distribution, Tweedie distribution, exponential distribution, etc.

For example,

1. Y is count (e.g. claim count): Poisson distribution
2. Y is binary (e.g., loss or no loss): Bernoulli distribution

$$\mathbf{g(\ E(Y)\ ) = g(\ \mu\ ) = X * \beta}$$

Link Functions

$X * \beta \in (-\infty,\infty)$

- Wrong: $E(Y) = \mu = X * \beta$

g() is required when the range of E(Y) differs from the range of $X * \beta$

- Correct: $g(E(Y)) = g(\mu) = X * \beta$

The domain of g() is matched to the range of E(Y).

The range of g() is matched to the range of $X * \beta$

For example,

1. Y is count (e.g. claim count): Poisson distribution

$E(Y) = \mu \in (0,\infty)$

g() can be log function: $g(\mu) = \ln(\mu) = X * \beta$

$g(\mu) \in (-\infty,\infty)$

Link Functions

2. Y is binary (e.g., loss or no loss): Bernoulli distribution

$E(Y) = \mu \in (0,1)$

g() can be logit function: $g(\mu) = \ln(\mu / (1-\mu)) = X * \beta$

or

g() can be Inverse CDF: $g(\mu)$ = Inverse of Normal CDF$(\mu) = X * \beta$

or

g() can be Complementary log-log function: $g(\mu) = \log(-\log(1-\mu)) = X * \beta$

$g(\mu) \in (-\infty,\infty)$

$$\mathbf{g(E(Y)) = g(\mu) = X * \beta}$$

R Function in stats package

"R function glm() is used to fit generalized linear models, specified by giving a symbolic description of the linear predictor and a description of the error distribution."

Inputs: glm(formula, family, data, …)

Outputs: coefficients, p values, residuals, fitted values, summary, …

$$\mathbf{g(E(Y)) = g(} \mu \mathbf{) = X *} \beta$$

Simulated Data

This data set is simulated. It records the numbers of personal auto claims incurred in 2015, the numbers of insured autos, the policyholders' ages, and their family sizes by policy level.

Variables	Descriptions
clm	Claim Counts
num_car	Number of Insured Personal Auto
age	Age of Policyholders
familiy_size	Family Size of Policyholders

```
rootDir<-"/Volumes/LEMMARIL/RPM Workshop/R Workshop/Github/rpm2016/"
clm_cnt <- read.csv(file=paste0(rootDir,"11_GLMs.csv"))
```

Summary and Graphs

```
dim(clm_cnt)
```

```
## [1] 240   4
```

```
colnames(clm_cnt)
```

```
## [1] "clm"          "age"          "num_car"      "family_size"
```

```
summary(clm_cnt)
```

```
##       clm              age            num_car        family_size
##  Min.   :0.0000   Min.   :15.0   Min.   :1.000   Min.   :1.00
##  1st Qu.:0.0000   1st Qu.:24.0   1st Qu.:1.000   1st Qu.:1.00
##  Median :1.0000   Median :44.5   Median :1.000   Median :1.00
##  Mean   :0.6833   Mean   :44.2   Mean   :1.417   Mean   :1.55
##  3rd Qu.:1.0000   3rd Qu.:61.0   3rd Qu.:2.000   3rd Qu.:2.00
##  Max.   :2.0000   Max.   :74.0   Max.   :4.000   Max.   :4.00
```

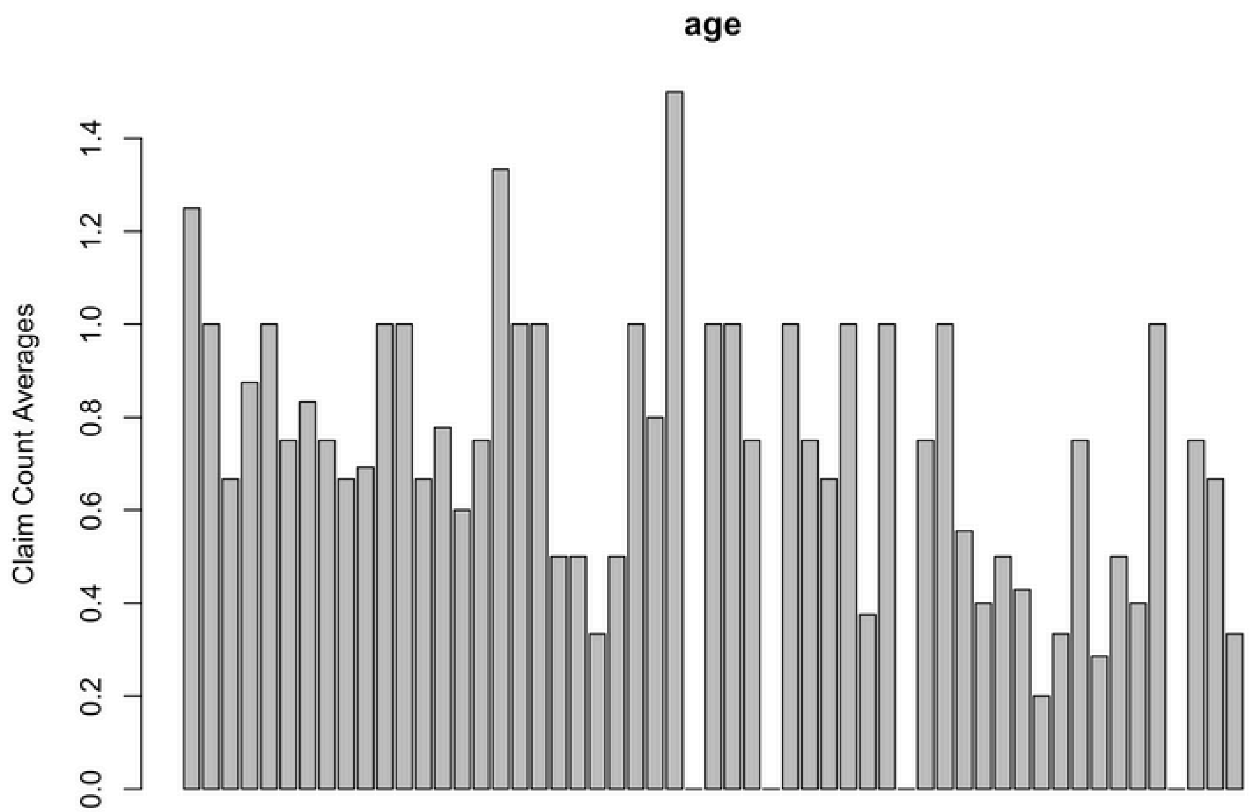
Summary and Graphs

```
apply(clm_cnt,2,table)
```

```
## $clm
##
##   0   1   2
##  81 154   5
##
## $age
##
## 15 16 17 18 19 20 21 22 23 24 26 28 29 30 31 33 34 35 36 37 38 39 40 41 42
##  4  6  3  8  8  4  6  4  6 13  2  2  3  9  5  4  3  3  3  4  2  3  4  3  5
## 43 44 45 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 66 67 68 69
##  2  1  2  4  4  1  1  4  3  2  8  5  1  8  1  9  5  4  7  5  6  4  7  2  5
## 70 71 72 73 74
##  4  2  4  3  9
##
## $num_car
##
##   1   2   3   4
## 160  63  14   3
##
## $family_size
##
##   1   2   3   4
## 142  68  26   4
```

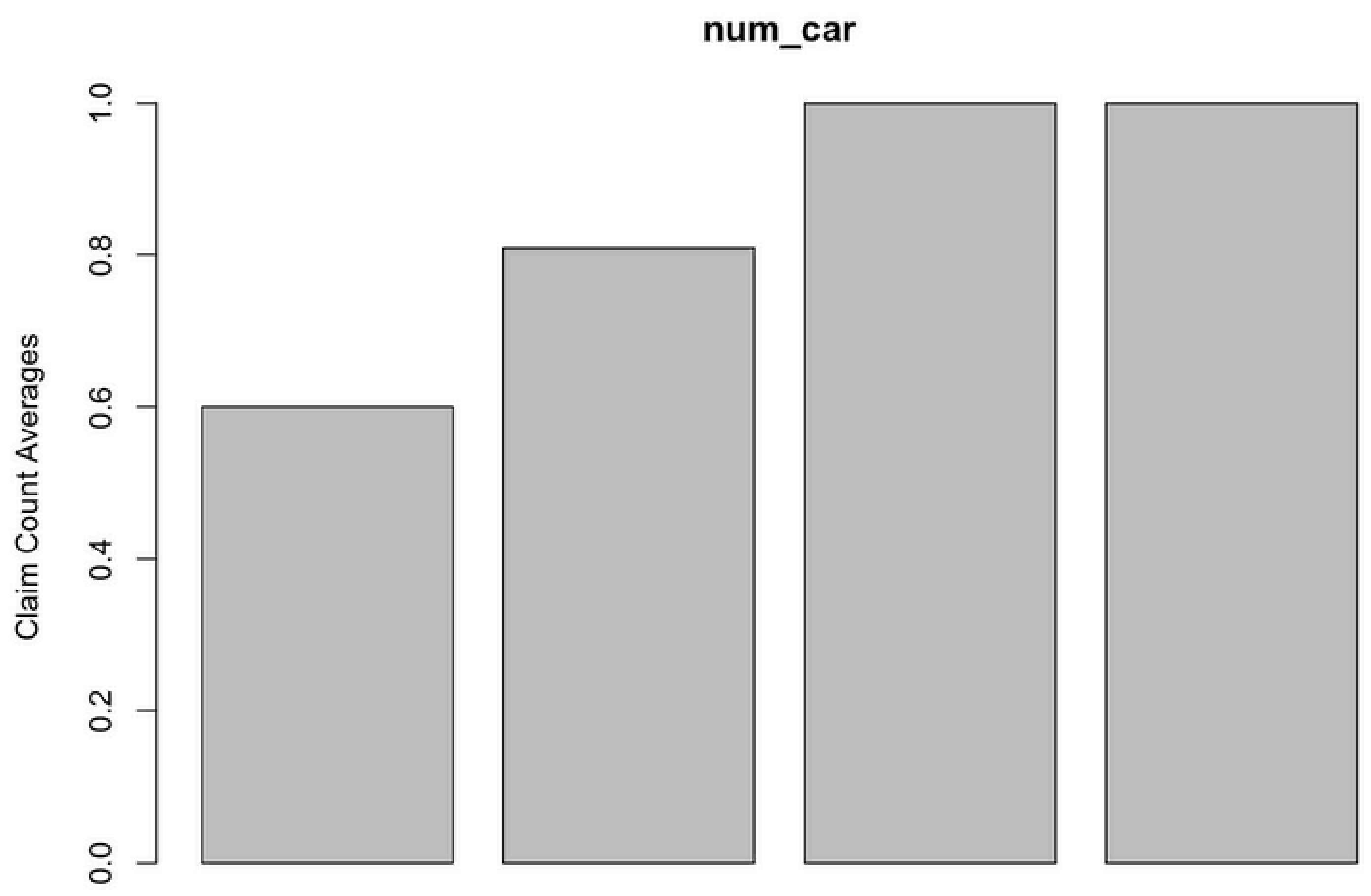
Summary and Graphs

```
avg<-function(x) {
  data<-aggregate(clm_cnt[,"clm"],by=list(clm_cnt[,x]),FUN=mean)
  barplot(data[,2],main=x,xlab=x,ylab="Claim Count Averages")
}
avg(x="age")
```



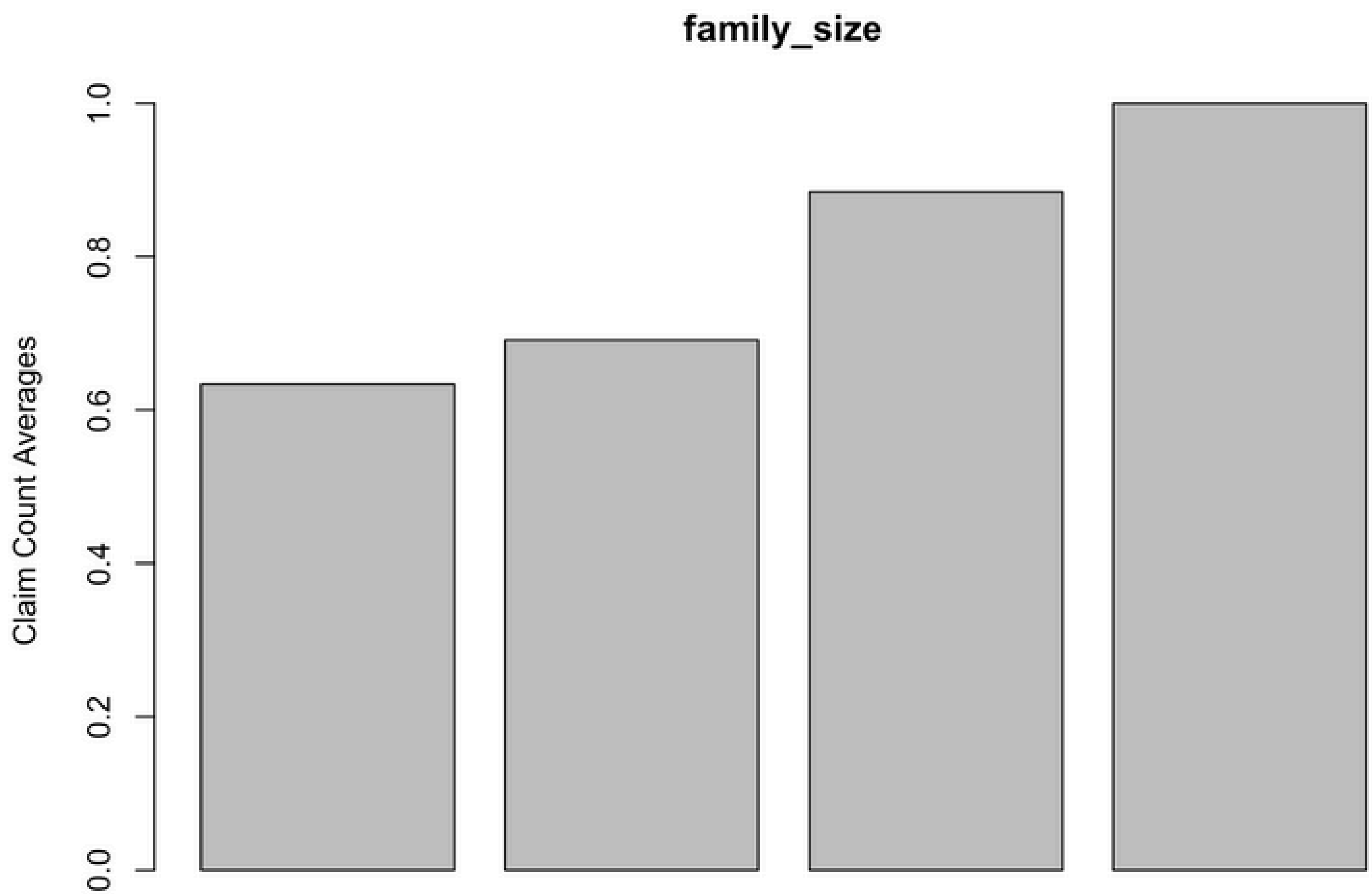
Summary and Graphs

```
avg(x="num_car")
```



Summary and Graphs

```
avg(x="family_size")
```



Summary and Graphs

```
cor(clm_cnt[,-1])
```

```
##                   age    num_car family_size
## age         1.0000000 0.10784077  0.10492571
## num_car     0.1078408 1.00000000  0.02524064
## family_size 0.1049257 0.02524064  1.00000000
```

Choose distribution and link function

1. Y is count (e.g. claim count): Poisson distribution
   log: $g(\mu) = \log(\mu) = X * \beta$

```
poisson_reg <- glm(clm~ age+num_car+family_size,   data = clm_cnt,   family = poisson)
summary(poisson_reg)
```

```
## Call:
## glm(formula = clm ~ age + num_car + family_size, family = poisson,
##     data = clm_cnt)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2860  -0.9268   0.1221   0.3887   1.8754
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.504041   0.275891  -1.827  0.06771 .
## age         -0.013699   0.004294  -3.190  0.00142 **
## num_car      0.277694   0.105531   2.631  0.00850 **
## family_size  0.182568   0.098658   1.851  0.06424 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 138.76  on 239  degrees of freedom
## Residual deviance: 121.24  on 236  degrees of freedom
## AIC: 450.3
##
## Number of Fisher Scoring iterations: 5
```

Choose distribution and link function

1. Y is count (e.g. claim count): Poisson distribution

   log: $g(\mu) = \log(\mu) = X * \beta$

```
coef(poisson_reg)
```

```
## (Intercept)         age     num_car family_size
##  -0.5040411  -0.0136992   0.2776940   0.1825675
```

```
str(poisson_reg)
```

```
## List of 30
##  $ coefficients     : Named num [1:4] -0.504 -0.0137 0.2777 0.1826
##   ..- attr(*, "names")= chr [1:4] "(Intercept)" "age" "num_car" "family_size"
##  $ residuals        : Named num [1:240] 0.337 0.642 -1 0.642 0.037 ...
##   ..- attr(*, "names")= chr [1:240] "1" "2" "3" "4" ...
##  $ fitted.values    : Named num [1:240] 0.748 0.609 0.609 0.609 0.964 ...
##   ..- attr(*, "names")= chr [1:240] "1" "2" "3" "4" ...
##  $ effects          : Named num [1:240] 4.192 -2.739 2.625 1.851 -0.133 ...
##   ..- attr(*, "names")= chr [1:240] "(Intercept)" "age" "num_car" "family_size" ...
##  $ R                : num [1:4, 1:4] -12.8 0 0 0 -513.7 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "(Intercept)" "age" "num_car" "family_size"
##   .. ..$ : chr [1:4] "(Intercept)" "age" "num_car" "family_size"
##  $ rank             : int 4
##  $ qr               :List of 5
##   ..$ qr   : num [1:240, 1:4] -12.8063 0.0609 0.0609 0.0609 0.0767 ...
##   .. ..- attr(*, "dimnames")=List of 2
```

Choose distribution and link function

1. Y is count (e.g. claim count): Poisson distribution

   log: g(μ) = log(μ) = X * β

```r
head(poisson_reg$residuals)
```

```
##          1          2           3          4          5          6
## 0.33691566 0.64189857 -1.00000000 0.64189857 0.03702878 2.32909207
```

```r
head(poisson_reg$fitted.values)
```

```
##         1         2         3         4         5         6
## 0.7479903 0.6090510 0.6090510 0.6090510 0.9642934 0.6007644
```

```r
head(poisson_reg$data)
```

```
##   clm age num_car family_size
## 1   1  18       1           1
## 2   1  33       1           1
## 3   0  33       1           1
## 4   1  33       1           1
## 5   1  40       3           1
## 6   2  34       1           1
```

Choose distribution and link function

1. Y is count (e.g. claim count): Poisson distribution

   log: $g(\mu) = \log(\mu) = X * \beta$

```
poisson_reg2 <- glm(clm~ age+num_car,   data = clm_cnt,   family = poisson)
summary(poisson_reg2)
```

```
##
## Call:
## glm(formula = clm ~ age + num_car, family = poisson, data = clm_cnt)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2997  -0.9575   0.1825   0.3976   1.7579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.24811    0.23422  -1.059  0.28947
## age         -0.01284    0.00425  -3.022  0.00251 **
## num_car      0.27717    0.10406   2.663  0.00773 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 138.76  on 239  degrees of freedom
## Residual deviance: 124.50  on 237  degrees of freedom
## AIC: 451.57
##
## Number of Fisher Scoring iterations: 5
```

Alternative distribution and link function

```
summary(clm_cnt$clm)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0000  0.0000  1.0000  0.6833  1.0000  2.0000
```

```
table(clm_cnt$clm)
```

```
##
##   0   1   2
##  81 154   5
```

2. Y is binary (e.g., loss or no loss): Bernoulli distribution

logit: $g(\mu) = \ln(\mu / (1-\mu)) = X * \beta$

```
clm_cnt$clm_cap <- ifelse(clm_cnt$clm==0,0,1)
logistic_reg <- glm(clm_cap~ age+num_car+family_size,   data= clm_cnt,   family = binomial)
summary(logistic_reg)
```

2. Y is binary (e.g., loss or no loss): Bernoulli distribution

logit: $g(\mu) = \ln(\mu / (1-\mu)) = X * \beta$

```
##
## Call:
## glm(formula = clm_cap ~ age + num_car + family_size, family = binomial,
##     data = clm_cnt)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1227  -0.8019   0.3648   0.7832   1.9362
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.504757   0.631501  -0.799 0.424117
## age         -0.056771   0.009853  -5.762 8.32e-09 ***
## num_car      1.772574   0.374012   4.739 2.14e-06 ***
## family_size  0.998410   0.258047   3.869 0.000109 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 306.89  on 239  degrees of freedom
## Residual deviance: 232.63  on 236  degrees of freedom
## AIC: 240.63
##
## Number of Fisher Scoring iterations: 5
```

Model Selection

AIC = 2 * *k* - *2* * ln(L)

k = the number of parameter

L = the maximized value of the likelihood function of the model M, p(x|M, Beta)

```
AIC(poisson_reg)
```

```
## [1] 450.3045
```

```
AIC(poisson_reg2)
```

```
## [1] 451.5671
```

```
AIC(logistic_reg)
```

```
## [1] 240.6336
```

Model Selection

BIC = k * *ln(n) - 2* * ln(L)

k = the number of parameter

L = the maximized value of the likelihood function of the model M, p(x|M, Beta)

n = the number of observations

```
BIC(poisson_reg)
```

```
## [1] 464.2271
```

```
BIC(poisson_reg2)
```

```
## [1] 462.009
```

```
BIC(logistic_reg)
```

```
## [1] 254.5562
```

Reference

https://en.wikipedia.org/wiki/Generalized_linear_model#Link_function

https://stat.ethz.ch/R-manual/R-devel/library/stats/html/glm.html

Q&A