MSBA7002_Tutorial_2

Yutao DENG

2023-11-10

Contents

1 Formating Markdown

1.1 Equations

$$L(\beta|X) = \prod_{i=1}^{n} p_i^a (1-p_i)^b$$

$$log(L(\beta|X)) = \sum_{i=1}^{n} (a * log(p_i) + b * log(1-p_i))$$

$$X = \begin{cases} 1 & \text{true} \\ 0 & \text{false} \end{cases}$$

1.2 Itemize and Enumerate

For PDF output:

- 1111
- 222
- 33

1. AAA
2. BB
3. C

For HTML output: (remember to leave an empty line before the list)

- Item A
- Item B
  - Item B1

1. Item 1
2. Item 2
3. Item 3
4. Item 4
   - Item 3a
   - Item 3b

2 Package Management

2.1 Install packages in R

```
# Install pacman package
# install.packages("pacman")


# Install package from url or file
# packageurl <- "https://cran.r-project.org/src/contrib/Archive/DiscriMiner/DiscriMine
# install.packages(packageurl, repos=NULL, type="source")
```

```r
# Load pacman package
library("pacman")

# Detach pacman package
detach("package:pacman")
```

2.2 Using pacman

```r
# package list for today: pROC,caret,dplyr,e1071

#Fisrt import the pacman package
# library("pacman")
# p_load(pROC,caret,dplyr,e1071)
# p_unload(pROC,caret,dplyr,e1071)

#Without loading the pacman package
# detach("package:pacman")
pacman::p_load(ggcorrplot, corrplot, car, nnet, GGally,kableExtra,pROC,caret,dplyr,e1071
```

3 Data manipulation

3.1 Using dplyr

```r
# Load the data of smartphone
sp <- read.csv("smartphone.dat")

# colnames(sp) # without dplyr
sp %>%
  colnames()
```

```
## [1] "Age"    "Income" "Rating" "Group"
```

```r
# sum(is.na(sp)) # without dplyr
sp %>%
```

```r
  is.na() %>%
  sum()
```

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

```r
# select from sp data where Group is Med
sp %>%
  filter(Group=='Med')
```

```
##     Age Income Rating Group
## 1  52.0   77.6    3.4   Med
## 2  47.1   74.6    4.4   Med
## 3  49.5   77.0    4.9   Med
## 4  59.0   79.0    3.2   Med
## 5  45.6   74.5    5.7   Med
## 6  49.1   69.6    3.3   Med
## 7  43.3   78.5    4.5   Med
## 8  43.4   79.6    6.4   Med
## 9  55.9   78.8    4.6   Med
## 10 49.5   72.8    4.7   Med
## 11 50.8   76.4    4.2   Med
## 12 54.8   73.9    4.5   Med
## 13 53.9   72.8    4.4   Med
## 14 43.7   76.5    4.7   Med
## 15 44.1   72.9    4.2   Med
## 16 45.8   70.8    2.3   Med
## 17 43.6   78.6    5.5   Med
## 18 47.0   76.4    5.6   Med
```

```r
# select from sp data where Group is Med or Low
sp %>% filter(Group %in% c('Med', 'Low'))
```

```
##     Age Income Rating Group
```

```
## 1  52.0   77.6    3.4   Med
## 2  29.1   44.8    3.5   Low
## 3  47.1   74.6    4.4   Med
## 4  49.5   77.0    4.9   Med
## 5  59.0   79.0    3.2   Med
## 6  45.6   74.5    5.7   Med
## 7  49.1   69.6    3.3   Med
## 8  39.4   37.2    0.6   Low
## 9  34.5   40.2    2.5   Low
## 10 43.3   78.5    4.5   Med
## 11 43.4   79.6    6.4   Med
## 12 55.9   78.8    4.6   Med
## 13 49.5   72.8    4.7   Med
## 14 50.8   76.4    4.2   Med
## 15 54.8   73.9    4.5   Med
## 16 20.9   24.2    1.5   Low
## 17 53.9   72.8    4.4   Med
## 18 43.7   76.5    4.7   Med
## 19 44.1   72.9    4.2   Med
## 20 45.8   70.8    2.3   Med
## 21 43.6   78.6    5.5   Med
## 22 47.0   76.4    5.6   Med
```

```
# select some column by name
sp %>%
  dplyr::select('Age','Group')
```

```
##     Age Group
## 1  56.8
## 2  52.0   Med
## 3  29.1   Low
```

```
## 4  77.4    Hi
## 5  70.7    Hi
## 6  47.1   Med
## 7  44.3
## 8  33.9
## 9  49.5   Med
## 10 56.3
## 11 55.0
## 12 59.0   Med
## 13 45.6   Med
## 14 49.1   Med
## 15 54.8
## 16 33.1
## 17 56.5
## 18 39.4   Low
## 19 51.8
## 20 51.3
## 21 41.9
## 22 67.3    Hi
## 23 57.3
## 24 54.3
## 25 43.8
## 26 34.5   Low
## 27 43.3   Med
## 28 43.4
## 29 43.4   Med
## 30 41.8
## 31 44.1
## 32 48.6
## 33 59.0
```

```
## 34 55.9   Med
## 35 50.1
## 36 38.4
## 37 33.3
## 38 58.4
## 39 55.9
## 40 40.0
## 41 70.1    Hi
## 42 49.5   Med
## 43 50.8   Med
## 44 40.0
## 45 54.2
## 46 51.9
## 47 54.8   Med
## 48 38.8
## 49 20.9   Low
## 50 51.8
## 51 40.3
## 52 59.6
## 53 53.9   Med
## 54 54.7
## 55 43.7   Med
## 56 44.1   Med
## 57 47.9
## 58 47.7
## 59 64.3
## 60 61.9
## 61 51.4    Hi
## 62 44.1
## 63 40.3
```

```
## 64 45.8   Med
## 65 43.6   Med
## 66 66.8    Hi
## 67 47.9
## 68 35.2
## 69 27.2
## 70 51.8
## 71 60.9
## 72 49.8
## 73 40.0
## 74 47.0   Med
## 75 75.2    Hi
```

```
# select some column by index
sp %>%
  dplyr::select(c(2,4))
```

```
##    Income Group
## 1    84.2
## 2    77.6   Med
## 3    44.8   Low
## 4   110.6    Hi
## 5   125.5    Hi
## 6    74.6   Med
## 7    54.8
## 8    62.1
## 9    77.0   Med
## 10   91.2
## 11   88.5
## 12   79.0   Med
## 13   74.5   Med
```

```
## 14   69.6   Med
## 15   85.1
## 16   63.9
## 17  105.1
## 18   37.2   Low
## 19   52.2
## 20   91.2
## 21   59.0
## 22  124.8    Hi
## 23   94.8
## 24   88.9
## 25   66.5
## 26   40.2   Low
## 27   78.5   Med
## 28   62.6
## 29   79.6   Med
## 30   82.6
## 31   95.2
## 32   83.4
## 33   92.7
## 34   78.8   Med
## 35   60.9
## 36   48.6
## 37   67.3
## 38  101.6
## 39   89.2
## 40   55.7
## 41  115.4    Hi
## 42   72.8   Med
## 43   76.4   Med
```

```
## 44   64.5
## 45   88.7
## 46   90.6
## 47   73.9   Med
## 48   59.8
## 49   24.2   Low
## 50   95.1
## 51   83.0
## 52   95.9
## 53   72.8   Med
## 54   86.1
## 55   76.5   Med
## 56   72.9   Med
## 57  102.8
## 58   51.3
## 59   97.9
## 60  107.5
## 61  112.0    Hi
## 62   66.6
## 63   54.6
## 64   70.8   Med
## 65   78.6   Med
## 66  119.5    Hi
## 67   66.9
## 68   59.9
## 69   56.0
## 70   84.5
## 71   90.1
## 72   89.3
## 73   64.4
```

```
## 74   76.4   Med
## 75  117.7    Hi
```

```
# use select and filter at the same time
sp %>%
  dplyr::select(c(1,2,4)) %>%
  dplyr::filter(Group %in% c('Med', 'Low'))
```

```
##     Age Income Group
## 1  52.0   77.6   Med
## 2  29.1   44.8   Low
## 3  47.1   74.6   Med
## 4  49.5   77.0   Med
## 5  59.0   79.0   Med
## 6  45.6   74.5   Med
## 7  49.1   69.6   Med
## 8  39.4   37.2   Low
## 9  34.5   40.2   Low
## 10 43.3   78.5   Med
## 11 43.4   79.6   Med
## 12 55.9   78.8   Med
## 13 49.5   72.8   Med
## 14 50.8   76.4   Med
## 15 54.8   73.9   Med
## 16 20.9   24.2   Low
## 17 53.9   72.8   Med
## 18 43.7   76.5   Med
## 19 44.1   72.9   Med
## 20 45.8   70.8   Med
## 21 43.6   78.6   Med
## 22 47.0   76.4   Med
```

```r
# gourp by Group
grp_sp <- sp %>% group_by(Group)
sp
```

```
##     Age Income Rating Group
## 1  56.8   84.2    4.4
## 2  52.0   77.6    3.4   Med
## 3  29.1   44.8    3.5   Low
## 4  77.4  110.6    7.4    Hi
## 5  70.7  125.5    7.9    Hi
## 6  47.1   74.6    4.4   Med
## 7  44.3   54.8    3.3
## 8  33.9   62.1    4.0
## 9  49.5   77.0    4.9   Med
## 10 56.3   91.2    5.5
## 11 55.0   88.5    5.0
## 12 59.0   79.0    3.2   Med
## 13 45.6   74.5    5.7   Med
## 14 49.1   69.6    3.3   Med
## 15 54.8   85.1    5.5
## 16 33.1   63.9    4.4
## 17 56.5  105.1    7.1
## 18 39.4   37.2    0.6   Low
## 19 51.8   52.2    1.8
## 20 51.3   91.2    6.3
## 21 41.9   59.0    3.2
## 22 67.3  124.8    8.2    Hi
## 23 57.3   94.8    5.0
## 24 54.3   88.9    6.0
## 25 43.8   66.5    3.7
```

```
## 26 34.5   40.2    2.5   Low
## 27 43.3   78.5    4.5   Med
## 28 43.4   62.6    4.5
## 29 43.4   79.6    6.4   Med
## 30 41.8   82.6    5.9
## 31 44.1   95.2    6.6
## 32 48.6   83.4    4.4
## 33 59.0   92.7    5.1
## 34 55.9   78.8    4.6   Med
## 35 50.1   60.9    3.3
## 36 38.4   48.6    2.2
## 37 33.3   67.3    5.7
## 38 58.4  101.6    6.5
## 39 55.9   89.2    6.9
## 40 40.0   55.7    4.3
## 41 70.1  115.4    7.0    Hi
## 42 49.5   72.8    4.7   Med
## 43 50.8   76.4    4.2   Med
## 44 40.0   64.5    4.7
## 45 54.2   88.7    5.8
## 46 51.9   90.6    5.4
## 47 54.8   73.9    4.5   Med
## 48 38.8   59.8    4.4
## 49 20.9   24.2    1.5   Low
## 50 51.8   95.1    6.7
## 51 40.3   83.0    5.6
## 52 59.6   95.9    6.0
## 53 53.9   72.8    4.4   Med
## 54 54.7   86.1    5.6
## 55 43.7   76.5    4.7   Med
```

```
## 56 44.1   72.9    4.2   Med
## 57 47.9  102.8    7.4
## 58 47.7   51.3    2.0
## 59 64.3   97.9    5.4
## 60 61.9  107.5    5.8
## 61 51.4  112.0    8.8    Hi
## 62 44.1   66.6    4.6
## 63 40.3   54.6    2.6
## 64 45.8   70.8    2.3   Med
## 65 43.6   78.6    5.5   Med
## 66 66.8  119.5    8.1    Hi
## 67 47.9   66.9    5.3
## 68 35.2   59.9    3.8
## 69 27.2   56.0    3.4
## 70 51.8   84.5    5.3
## 71 60.9   90.1    5.0
## 72 49.8   89.3    6.1
## 73 40.0   64.4    4.7
## 74 47.0   76.4    5.6   Med
## 75 75.2  117.7    7.2    Hi
```

```
grp_sp
```

```
## # A tibble: 75 x 4
## # Groups:   Group [4]
##       Age Income Rating Group
##     <dbl>  <dbl>  <dbl> <chr>
##   1  56.8   84.2    4.4 ""
##   2  52     77.6    3.4 "Med"
##   3  29.1   44.8    3.5 "Low"
##   4  77.4  111.     7.4 "Hi"
```

```
## 5  70.7  126.     7.9 "Hi"
## 6  47.1   74.6    4.4 "Med"
## 7  44.3   54.8    3.3 ""
## 8  33.9   62.1    4   ""
## 9  49.5   77      4.9 "Med"
## 10 56.3   91.2    5.5 ""
## # i 65 more rows
```

```r
grp_sp %>% summarise(sum(Income))
```

```
## # A tibble: 4 x 2
##   Group `sum(Income)`
##   <chr>         <dbl>
## 1 ""            3583.
## 2 "Hi"           826.
## 3 "Low"          146.
## 4 "Med"         1360.
```

4 Formating Results

4.1 Using kableExtra

```r
# Basic HTML table
kbl(head(sp))
```

Age	Income	Rating	Group
56.8	84.2	4.4	
52.0	77.6	3.4	Med
29.1	44.8	3.5	Low
77.4	110.6	7.4	Hi
70.7	125.5	7.9	Hi
47.1	74.6	4.4	Med

```r
# bootstrap theme
sp %>%
  head() %>%
  kbl() %>%
  kable_styling()
```

Age	Income	Rating	Group
56.8	84.2	4.4	
52.0	77.6	3.4	Med
29.1	44.8	3.5	Low
77.4	110.6	7.4	Hi
70.7	125.5	7.9	Hi
47.1	74.6	4.4	Med

Table 1: Smartphone data table

Age	Income	Rating	Group
56.8	84.2	4.4	
52.0	77.6	3.4	Med
29.1	44.8	3.5	Low
77.4	110.6	7.4	Hi
70.7	125.5	7.9	Hi
47.1	74.6	4.4	Med

```
# add caption
sp %>%
  head() %>%
  kbl(caption = "Smartphone data table") %>%
  kable_styling()
```

```
# Try other themes by replacing the last line
# kable_paper, kable_classic, kable_classic_2, kable_minimal, kable_material and kable
sp %>%
  head() %>%
  kbl(caption = "Smartphone data table", centering = T, align = c('r', 'r', 'r')) %>%
  kable_classic(full_width = T, html_font = "Cambria")
```

```
# Add more details
sp %>%
```

Table 2: Smartphone data table

Age	Income	Rating	Group
56.8	84.2	4.4	
52.0	77.6	3.4	Med
29.1	44.8	3.5	Low
77.4	110.6	7.4	Hi
70.7	125.5	7.9	Hi
47.1	74.6	4.4	Med

Table 3: Smartphone data table

AIR			Categorial
Age	Income	Rating	Group
1-3			
56.8	84.2	4.4	
52.0	77.6	3.4	Med
29.1	44.8	3.5	Low
4-6			
77.4	110.6	7.4	Hi
70.7	125.5	7.9	Hi
47.1	74.6	4.4	Med

```
  head() %>%
  kbl(caption = "Smartphone data table", centering = T, align = c('r', 'r', 'r')) %>%
  kable_styling() %>%
  add_header_above(c(" AIR " = 3, "Categorial" = 1))%>%
  pack_rows(index = c("1-3" = 3, "4-6" = 3))
```

```
# For pdf output
# , format="latex"
sp %>%
  head() %>%
  kbl(booktabs = TRUE, format="latex", escape = F, align = c('r', 'r', 'r'),
      col.names = c("","$\\Pi$", "$\\hat{a}$", "$R^2$")) %>%
  kable_styling(latex_options = "striped", full_width = TRUE, font_size = 12)
```

	Π	$\hat{a}$	R^2
56.8	84.2	4.4	
52.0	77.6	3.4	Med
29.1	44.8	3.5	Low
77.4	110.6	7.4	Hi
70.7	125.5	7.9	Hi
47.1	74.6	4.4	Med

4.2 Figures

4.2.1 Figures for EDA

```r
# Use the iris data in GGally
iris%>%colnames()
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

```r
# # scatter plots
# # Use geom_point
# iris%>%
#   filter(Species=="setosa")%>%
#   ggplot(aes(x = Petal.Length, y = Petal.Width)) +
#   geom_point()


# iris%>%
#   filter(Species=="setosa")%>%
#   dim()
#
# # Add transparency to show all points
# iris%>%
#   filter(Species=="setosa")%>%
#   ggplot(aes(x = Petal.Length, y = Petal.Width)) +
#   geom_point(alpha = 0.3, size = 2.0)
#
# # Also can use geom_jitter to add random shift
# iris%>%
#   filter(Species=="setosa")%>%
#   ggplot(aes(x = Petal.Length, y = Petal.Width)) +
#   geom_jitter(width=0.05, height=0.05)
#
#
# # Set colors for each group
# iris%>%
```

```
#   ggplot(aes(x = Petal.Length,
#              y = Petal.Width,
#              color = Species)
#          ) +
#   geom_jitter(width=0.05, height=0.05)
#
# # Set different size by the sepal.length
# iris%>%
#   ggplot(aes(x = Petal.Length,
#              y = Petal.Width,
#              color = Species,
#              size = Sepal.Length)
#          ) +
#   geom_jitter(width=0.05, height=0.05, alpha = 0.4)
```

```
# # ggpairs: generalized pairs plot
# ggpairs(iris)
#
# # select columns and color by group
# iris %>%
#   ggpairs(columns = 1:4,      # Columns
#         aes(color = Species,  # Color by group (cat. variable)
#             alpha = 0.5))     # Transparency
#
# # Figure for correlation matrix
# # Use ggcorrplot
# iris[,1:4]%>%
#   cor()%>%
#   ggcorrplot(hc.order=TRUE)
#
```

```r
# # Use corrplot
# iris[,1:4]%>%
#   cor()%>%
#   corrplot()
```

4.2.2 Figures for linear models

```r
# Recall the linear regression in last tutorial using smartphone data
# lm_fit <- lm(Rating ~ Income, data = sp)

# sp%>%
#   ggplot(aes(x = Rating, y = Income, color = Group)) +
#   geom_point() +
#   geom_smooth(method = "lm", formula = "y~x")
```

5 Logistic regression

```r
# Use the titanic data
tit <- read.csv("titanicpassengers-bbm.dat")
str(tit)
```

```
## 'data.frame':    1309 obs. of  10 variables:
##  $ Name                  : chr  "Allen, Miss. Elisabeth Walton" "Allison, Master. Hudso
##  $ Survived              : chr  "Yes" "Yes" "No" "No" ...
##  $ Passenger.Class       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Sex                   : chr  "female" "male" "female" "male" ...
##  $ Age                   : num  29 0.917 2 30 25 ...
##  $ Siblings.and.Spouses: int  0 1 1 1 1 0 1 0 2 0 ...
##  $ Parents.and.Children: int  0 2 2 2 2 0 0 0 0 0 ...
##  $ Fare                  : num  211 152 152 152 152 ...
##  $ Port                  : chr  "S" "S" "S" "S" ...
##  $ Home...Destination   : chr  "St Louis, MO" "Montreal, PQ / Chesterville, ON" "Montr
```

```r
tit$Passenger.Class <- factor(tit$Passenger.Class)
tit$Port <- factor(tit$Port, levels = c("C","Q","S"))
tit$Survived <- factor(tit$Survived)
# glm() generalized linear model
## it requires y as a factor varaible
```

```r
# check the missing value
apply(tit, 2, function(x) sum(is.na(x)))
```

```
##                 Name             Survived      Passenger.Class
##                    0                    0                    0
##                  Sex                  Age Siblings.and.Spouses
##                    0                  263                    0
## Parents.and.Children                 Fare                 Port
##                    0                    1                    2
##    Home...Destination
##                    0
```

```r
# Remove the rolls with missing values
# since we want to keep the age column
tit.comp <- na.omit(tit)
```

```r
# fit a logistic regression with all variables without name and Home
# use glm()
fit_1.1 <- glm(Survived ~., data = tit.comp[,-c(1,10)], family = binomial(logit))
summary(fit_1.1)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(logit), data = tit.comp[,
##     -c(1, 10)])
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)           4.2571086  0.4298652   9.903  < 2e-16 ***
## Passenger.Class2     -1.1093557  0.2692636  -4.120 3.79e-05 ***
## Passenger.Class3     -2.0519804  0.2777922  -7.387 1.50e-13 ***
## Sexmale              -2.6128646  0.1795723 -14.550  < 2e-16 ***
## Age                  -0.0381233  0.0067240  -5.670 1.43e-08 ***
## Siblings.and.Spouses -0.3512240  0.1086969  -3.231  0.00123 **
## Parents.and.Children  0.0512518  0.1041801   0.492  0.62275
## Fare                  0.0002743  0.0019752   0.139  0.88954
## PortQ                -1.4485524  0.4460048  -3.248  0.00116 **
## PortS                -0.6620573  0.2154290  -3.073  0.00212 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1409.99  on 1042  degrees of freedom
## Residual deviance:  954.57  on 1033  degrees of freedom
## AIC: 974.57
##
## Number of Fisher Scoring iterations: 5
```

```r
# fit a logistic regression with all variables without name, Fare, Parents and Home
# use glm()
fit_1.2 <- glm(Survived ~., data = tit.comp[,-c(1,7,8,10)], family = binomial(logit))
summary(fit_1.2)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(logit), data = tit.comp[,
##     -c(1, 7, 8, 10)])
##
```

```
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)    
## (Intercept)             4.315656   0.382859  11.272  < 2e-16 ***
## Passenger.Class2       -1.126422   0.243779  -4.621 3.82e-06 ***
## Passenger.Class3       -2.069269   0.238929  -8.661  < 2e-16 ***
## Sexmale                -2.632629   0.176375 -14.926  < 2e-16 ***
## Age                    -0.038306   0.006712  -5.707 1.15e-08 ***
## Siblings.and.Spouses -0.332316   0.103047  -3.225 0.001260 ** 
## PortQ                  -1.471228   0.444588  -3.309 0.000936 ***
## PortS                  -0.668459   0.212694  -3.143 0.001673 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1409.99  on 1042  degrees of freedom
## Residual deviance:  954.88  on 1035  degrees of freedom
## AIC: 970.88
##
## Number of Fisher Scoring iterations: 5
```

```r
# Apply model selection by anova and Anova
anova_out <- anova(fit_1.1, fit_1.2, test = "Chisq")
Anova_out <- Anova(fit_1.1)
anova_out
```

```
## Analysis of Deviance Table
##
## Model 1: Survived ~ Passenger.Class + Sex + Age + Siblings.and.Spouses +
##     Parents.and.Children + Fare + Port
## Model 2: Survived ~ Passenger.Class + Sex + Age + Siblings.and.Spouses +
```

```
##     Port
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1033     954.57
## 2      1035     954.88 -2  -0.3119   0.8556
```

```r
# conduct the final model
fit_1_final <- glm(formula = Survived ~ Passenger.Class + Sex + Age + Siblings.and.Spous
    Port, family = binomial(logit), data = tit.comp[, -c(1, 10)])
summary(fit_1_final)
```

```
##
## Call:
## glm(formula = Survived ~ Passenger.Class + Sex + Age + Siblings.and.Spouses +
##     Port, family = binomial(logit), data = tit.comp[, -c(1, 10)])
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           4.315656   0.382859  11.272  < 2e-16 ***
## Passenger.Class2     -1.126422   0.243779  -4.621 3.82e-06 ***
## Passenger.Class3     -2.069269   0.238929  -8.661  < 2e-16 ***
## Sexmale              -2.632629   0.176375 -14.926  < 2e-16 ***
## Age                  -0.038306   0.006712  -5.707 1.15e-08 ***
## Siblings.and.Spouses -0.332316   0.103047  -3.225 0.001260 **
## PortQ                -1.471228   0.444588  -3.309 0.000936 ***
## PortS                -0.668459   0.212694  -3.143 0.001673 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1409.99  on 1042  degrees of freedom
```

```
## Residual deviance:  954.88  on 1035  degrees of freedom
## AIC: 970.88
##
## Number of Fisher Scoring iterations: 5
```

```r
# Do prediction


# Create your jack


jack <- tit.comp[1,]
jack[1,] <- c("Jack","",3,"male",17,0,"","","S","")
jack$Age <- jack$Age %>% as.numeric()
jack$Siblings.and.Spouses <- jack$Siblings.and.Spouses %>% as.numeric()


# use predict()
predict(fit_1_final, jack, type = "response")
```

```
##         1
## 0.1536967
```

```r
predict(fit_1_final, jack)
```

```
##         1
## -1.705897
```

```r
logodd = predict(fit_1_final, jack)
exp(logodd)/(exp(logodd)+1)
```

```
##         1
## 0.1536967
```

```r
# Draw the ROC curve for the final model


# roc_final <- roc(tit.comp$Survived, fit_1_final$fitted.values)


# # plot the roc curve with ggroc
```

```
# roc_final%>%
#   ggroc(colour = 'steelblue', size = 2) +
#   ggtitle(paste('ROC Curve ', '(AUC = ', round(roc_final$auc,4), ')'))
```

6 Multinomial Logistic Regression

```
# Y: n levels
## n-1 logistics regressions
## (n-1)*p parameters
```

6.1

```
# Use nnet to conduct Multinominal
# multinominal regression <=> single hidden layer neural network activated by sigmoid

library(nnet)
```

6.2 Nominal Response

```
# Use WVS data from {carData}

fit_mult <- multinom(poverty ~ ., data = WVS)
```

```
## # weights:  27 (16 variable)
## initial  value 5911.632725
## iter  10 value 5305.513352
## iter  20 value 5011.780398
## final  value 5011.083370
## converged
```

```
str(WVS)
```

```
## 'data.frame':    5381 obs. of  6 variables:
##  $ poverty : Ord.factor w/ 3 levels "Too Little"<"About Right"<..: 1 2 1 3 1 2 3 1 1
##  $ religion: Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...
```

```
##  $ degree  : Factor w/ 2 levels "no","yes": 1 1 1 2 2 1 1 1 1 1 ...
##  $ country : Factor w/ 4 levels "Australia","Norway",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ age     : int  44 40 36 25 39 80 48 32 74 30 ...
##  $ gender  : Factor w/ 2 levels "female","male": 2 1 1 1 2 1 1 2 1 2 ...
```

```r
summary(fit_mult)
```

```
## Call:
## multinom(formula = poverty ~ ., data = WVS)
##
## Coefficients:
##             (Intercept) religionyes degreeyes countryNorway countrySweden
## About Right  -0.8955989 -0.02123626 0.1989984     0.1996286   -0.09824553
## Too Much     -2.2741153  0.36527167 0.1075563    -1.6949273   -2.11205874
##              countryUSA         age gendermale
## About Right -0.03787903 0.008430301  0.2037626
## Too Much     0.88665014 0.015961608  0.1769822
##
## Std. Errors:
##             (Intercept) religionyes  degreeyes countryNorway countrySweden
## About Right   0.1199008  0.08900485 0.07563377    0.08256209    0.08747003
## Too Much      0.1671035  0.11883164 0.11335282    0.18640893    0.21719328
##             countryUSA         age gendermale
## About Right 0.08621725 0.001815380 0.06082448
## Too Much    0.09499034 0.002442526 0.08538729
##
## Residual Deviance: 10022.17
## AIC: 10054.17
```

```r
Anova(fit_mult)
```

```
## Analysis of Deviance Table (Type II tests)
##
```

```
## Response: poverty
##          LR Chisq Df Pr(>Chisq)
## religion    11.50  2   0.003183 **
## degree       6.95  2   0.030984 *
## country    613.92  6  < 2.2e-16 ***
## age         50.65  2  1.002e-11 ***
## gender      12.56  2   0.001873 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Create a new sample
newdata <- WVS[1,]
newdata[1,] <- c("", "yes", "yes", "USA", 18 , "female")
newdata$age <- as.numeric(newdata$age)
```

```r
# Prediction
predict(fit_mult,newdata, type = "class")
```

```
## [1] Too Little
## Levels: Too Little About Right Too Much
```

```r
predict(fit_mult,newdata, type = "prob")
```

```
##  Too Little About Right    Too Much
##   0.4806182    0.2627237   0.2566581
```

6.3 Ordinal Response

```r
pacman::p_load(MASS)
```

```r
# use polr to apply ordinal logistic
fit_ord <- polr(poverty ~ ., data = WVS)
summary(fit_ord)
```

```
##
## Re-fitting to get Hessian

## Call:
```

```
## polr(formula = poverty ~ ., data = WVS)
##
## Coefficients:
##                   Value Std. Error t value
## religionyes     0.17973   0.077346   2.324
## degreeyes       0.14092   0.066193   2.129
## countryNorway -0.32235   0.073766  -4.370
## countrySweden -0.60330   0.079494  -7.589
## countryUSA      0.61777   0.070665   8.742
## age             0.01114   0.001561   7.139
## gendermale      0.17637   0.052972   3.329
##
## Intercepts:
##                         Value   Std. Error t value
## Too Little|About Right  0.7298  0.1041     7.0128
## About Right|Too Much    2.5325  0.1103    22.9496
##
## Residual Deviance: 10402.59
## AIC: 10420.59
```

```r
Anova(fit_ord)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: poverty
##          LR Chisq Df Pr(>Chisq)
## religion    5.434  1   0.019753 *
## degree      4.518  1   0.033542 *
## country   250.881  3  < 2.2e-16 ***
## age        51.120  1   8.69e-13 ***
## gender     11.096  1   0.000865 ***
```

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

```r
predict(fit_ord, newdata, type = "class")
```

```
## [1] About Right
## Levels: Too Little About Right Too Much
```

```r
predict(fit_ord, newdata, type = "prob")
```

```
##  Too Little About Right    Too Much
##   0.3991052   0.4020485   0.1988464
```