# Statistics and Data Analysis

# R Programming and Logistic Regression

Ling-Chieh Kung

Department of Information Management
National Taiwan University

# Road map

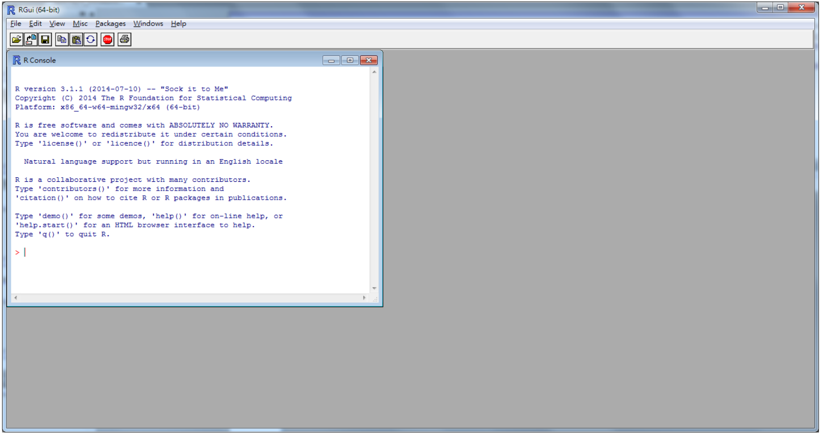
- **The R programming language**.
- Regression in R.
- Logistic regression.

# The R programming language

- **R** is a programming language for statistical computing and graphics.
- R is open source.
- R is powerful and flexible.
  - It is fast.
  - Most statistical methods have been implemented as packages.
  - One may write her own R programs to complete her own task.
- `http://www.r-project.org/`.
- To download, go to `http://cran.csie.ntu.edu.tw/`, choose your platform, then choose the suggested one (the current version is 3.2.3).

# The programming environment

- When you run R, you should see this:



# Try it!

- Type some mathematical expressions!

```
> 1 + 2
[1] 3
> 6 * 9
[1] 54
> 3 * (2 + 3) / 4
[1] 3.75
> log(2.718)
[1] 0.9998963
> 10 ^ 3
[1] 1000
> sqrt(25)
[1] 5
```

# Let's do statistics

- A wholesaler has 440 customers in Portugal:
  - 298 are "horeca"s (hotel/restaurant/café).
  - 142 are retails.
- These customers locate at different regions:
  - Lisbon: 77.
  - Oporto: 47.
  - Others: 316.
- Data source: `http://archive.ics.uci.edu/ml/datasets/Wholesale+customers`.

# Let's do statistics

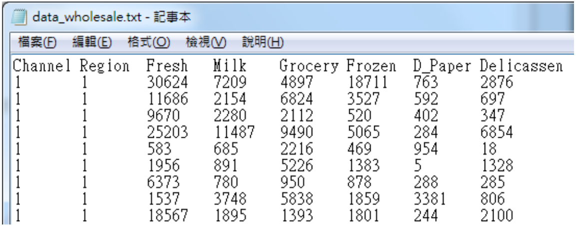
- The data:

| Channel | Label | Fresh | Milk | Grocery | Frozen | D. & P. | Deli. |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 30624 | 7209 | 4897 | 18711 | 763 | 2876 |
| 1 | 1 | 11686 | 2154 | 6824 | 3527 | 592 | 697 |
| | | | | ⋮ | | | |
| 2 | 3 | 14531 | 15488 | 30243 | 437 | 14841 | 1867 |

- The wholesaler records the annual amount each customer spends on six product categories:
  - Fresh, milk, grocery, frozen, detergents and paper, and delicatessen.
  - Amounts have been scaled to be based on "monetary unit."
- Channel: hotel/restaurant/café = 1, retailer = 2.
- Region: Lisbon = 1, Oporto = 2, others = 3.

# Data in a TXT file

- The data are provided in an MS Excel worksheet "wholesale."
- Let's **copy and paste** the data to a TXT file "wholesale.txt."
- Copying data from Excel and pasting them to a TXT file will make data in columns **separated by tabs**.



data_wholesale.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

| Channel | Region | Fresh | Milk | Grocery | Frozen | D_Paper | Delicassen |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 30624 | 7209 | 4897 | 18711 | 763 | 2876 |
| 1 | 1 | 11686 | 2154 | 6824 | 3527 | 592 | 697 |
| 1 | 1 | 9670 | 2280 | 2112 | 520 | 402 | 347 |
| 1 | 1 | 25203 | 11487 | 9490 | 5065 | 284 | 6854 |
| 1 | 1 | 583 | 685 | 2216 | 469 | 954 | 18 |
| 1 | 1 | 1956 | 891 | 5226 | 1383 | 5 | 1328 |
| 1 | 1 | 6373 | 780 | 950 | 878 | 288 | 285 |
| 1 | 1 | 1537 | 3748 | 5838 | 1859 | 3381 | 806 |
| 1 | 1 | 18567 | 1895 | 1393 | 1801 | 244 | 2100 |

- DO NOT modify anything after pasting even if data are not aligned perfectly. Just copy and paste.

# Reading data from a TXT file

- Let's put the TXT file to your **work directory**.
  - A file should be put in the work directory for R to read data from it.[1]
- To find the default work directory:[2]

```
> getwd()
[1] "C:/Users/user/Documents"
```

- To **read** the data into R, we execute:

```
> W <- read.table("wholesale.txt", header = TRUE)
```

  - `W` is a **data frame** that stores the data.
  - `<-` assigns the right-hand-side values to the variable at its left.

[1] Or one may use `setwd()` to choose an existing folder as the work directory.
[2] The work directory on your computer may be different from mine.

# Browsing data

- To browse the data stored in a data frame:

```
> W
> head(W)
> tail(W)
```

- To extract a row or a column:

```
> W[1, ]
> W$Channel
> W[, 1]
```

- What is this?

```
> W[1, 2]
```

# Basic statistics

- The **mean**, **median**, **max**, and **min** expenditure on milk:

```
> mean(W$Milk)
> median(W$Milk)
> max(W$Milk)
> min(W$Milk)
```

- The **sample standard deviation** of expenditure on milk:

```
> sd(W$Milk)
```

- **Counting**:

```
> length(W[1, ])
> length(W[, 1])
```

# Basic statistics

- **Correlation coefficient**:

```
> cor(W$Milk, W$Grocery)
```

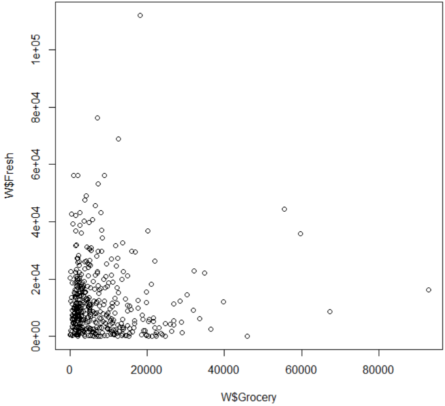
- In fact, you may simply do:

```
> W2 <- W[, 3:8]
> cor(W2)
```

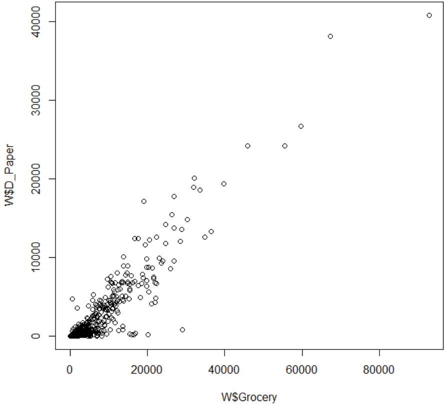
  - `3:8` is a vector $(3, 4, 5, 6, 7, 8)$.
  - `W[, 3:8]` is the third to the eighth columns of `W`.
  - `cor(W2)` is the **correlation matrix** for pairwise correlation coefficients among all columns of `W2`.

# Basic graphs: Scatter plots

```
> plot(W$Grocery, W$Fresh)
```

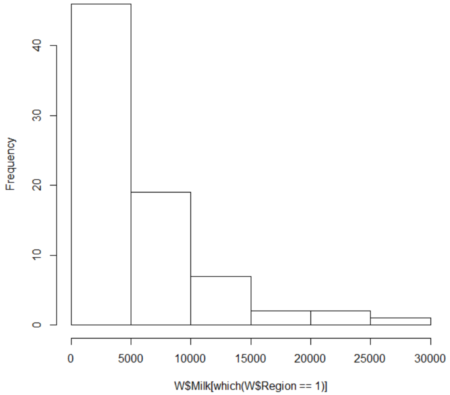



```
> plot(W$Grocery, W$D_Paper)
```



# Basic graphs: histograms

```
> hist(W$Milk[which(W$Region == 1)])
```



# Writing scripts in a file

- It is suggested to **write scripts** (codes) in a **file**.
  - This makes the codes easily modified and reusable.
  - Multiple statements may be executed at the same time.
  - These codes can be stored for future uses.
- To do so, open a new script file in R and then write codes line by line.
  - Execute a line of codes by pressing "**Ctrl + R**" in Windows or "**Command + return (enter)**" in Mac.
  - Select **multiple lines of codes** and then execute all of them together in the same way.
- In your file, put **comments** (personal notes of your program) after **#**. Characters after # will be ignored when executing a line of codes.
- The saved .R files can be edit by any **plain text editor**.
  - E.g., Notepad in Windows.

# Road map

- The R programming language.
- **Regression in R**.
- Logistic regression.

# Regression in R

- Let's do regression in R. First, let's load the data:
  - Copy all the data in the MS Excel worksheet "bike_day."
  - Paste them into a TXT file with "bike.txt" as the file name.
  - Put the file in the work directory.
  - Execute

    ```
    B <- read.table("bike_day.txt", header = TRUE)
    ```

- Take a look at B:

  ```
  head(B)
  mean(B$cnt)
  cor(B$cnt, B$temp)
  hist(B$cnt)
  ```

- Try them!

  ```
  pairs(B)
  pairs(B[, 10:16])
  ```

# Simple regression

- Let's build a **simple regression** model by using the function `lm()`:

```
fit <- lm(B$cnt ~ B$instant)
summary(fit)
```

  - Put the dependent variable **before** the ~ operator.
  - Put the independent variable **after** the ~ operator.
- We will obtain the regression report:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2392.9613   111.6133   21.44   <2e-16 ***
B$instant      5.7688     0.2642   21.84   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1507 on 729 degrees of freedom
Multiple R-squared:  0.3954,    Adjusted R-squared:  0.3946
F-statistic: 476.8 on 1 and 729 DF,  p-value: < 2.2e-16
```

# Multiple regression

- Let's **add more variables** using the + operator:

```
fit <- lm(B$cnt ~ B$instant + B$workingday + B$temp)
summary(fit)
```

- The regression report:

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -280.3863   138.8325   -2.02   0.0438 *
B$instant       5.0197     0.1925   26.07   <2e-16 ***
B$workingday  145.3731    86.5121    1.68   0.0933 .
B$temp        140.2238     5.4246   25.85   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1086 on 727 degrees of freedom
Multiple R-squared:  0.6871,    Adjusted R-squared:  0.6858
F-statistic: 532.1 on 3 and 727 DF,  p-value: < 2.2e-16
```

# Interaction

- Let's consider **interaction** using the * operator:

```
fit <- lm(B$cnt ~ B$instant + B$workingday * B$temp)
summary(fit)
```

- The regression report:

```
Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           -631.776    204.732  -3.086  0.00211 **
B$instant                5.026      0.192  26.183  < 2e-16 ***
B$workingday           675.120    243.232   2.776  0.00565 **
B$temp                 157.912      9.323  16.938  < 2e-16 ***
B$workingday:B$temp    -26.471     11.364  -2.329  0.02012 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1083 on 726 degrees of freedom
Multiple R-squared:  0.6894,    Adjusted R-squared:  0.6877
F-statistic: 402.9 on 4 and 726 DF,  p-value: < 2.2e-16
```

# Qualitative variables

- Let's add a non-binary **qualitative variable** (in a **wrong** way):

```
fit <- lm(B$cnt ~ B$instant + B$workingday * B$temp + B$season)
summary(fit)
```

- The regression report:

```
Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          -628.7340   208.7156  -3.012  0.00268 **
B$instant               5.0324     0.2085  24.141  < 2e-16 ***
B$workingday          675.0576   243.3996   2.773  0.00569 **
B$temp                158.0409     9.4807  16.670  < 2e-16 ***
B$season               -3.1710    41.5623  -0.076  0.93921
B$workingday:B$temp   -26.4682    11.3722  -2.327  0.02022 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1083 on 725 degrees of freedom
Multiple R-squared:  0.6894,    Adjusted R-squared:  0.6873
F-statistic: 321.9 on 5 and 725 DF,  p-value: < 2.2e-16
```

# Qualitative variables

- To correctly include a qualitative variable, use the function `factor()`:

```
fit <- lm(B$cnt ~ B$instant + B$workingday * B$temp + factor(B$season))
summary(fit)
```

  - `factor()` tells the R program to interpret those values as categories even if they are numbers.
  - If the values are already non-numeric, there is no need to use `factor()`.
- Let's read the regression report.

# Qualitative variables

- The regression report:[3]

```
Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        -749.4834   209.3085  -3.581 0.000366 ***
B$instant             5.1296     0.2015  25.459  < 2e-16 ***
B$workingday        632.4411   233.8650   2.704 0.007006 **
B$temp              146.5942    11.7999  12.423  < 2e-16 ***
factor(B$season)2   827.2798   143.1463   5.779 1.12e-08 ***
factor(B$season)3   142.7658   188.6595   0.757 0.449454
factor(B$season)4   272.6144   126.7112   2.151 0.031770 *
B$workingday:B$temp -24.5086    10.9264  -2.243 0.025195 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1041 on 723 degrees of freedom
Multiple R-squared:  0.7142,    Adjusted R-squared:  0.7115
F-statistic: 258.2 on 7 and 723 DF,  p-value: < 2.2e-16
```

[3] To change the reference level, use `relevel()`.

# Transformation: method 1

- To add $temp^2$, there are two ways:

```
tempSq <- B$temp^2
fit <- lm(B$cnt ~ B$instant + B$workingday * (B$temp + tempSq))
summary(fit)
```

- The regression report:

```
Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         -3313.2904   462.5027  -7.164 1.93e-12 ***
B$instant               4.7928     0.1874  25.576  < 2e-16 ***
B$workingday         1934.5264   578.2195   3.346 0.000863 ***
B$temp                482.5310    50.6541   9.526  < 2e-16 ***
tempSq                 -8.1197     1.2489  -6.501 1.48e-10 ***
B$workingday:B$temp  -180.0186    62.5810  -2.877 0.004138 **
B$workingday:tempSq     3.9116     1.5382   2.543 0.011200 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

# Transformation: method 2

- Alternatively, we may create the new variable as a **new column** in the MS Excel worksheet.
- Then copy and paste to update the content in the TXT file.
- Execute `read.table()` again to update the data frame `B`.
- Finally, redo `lm()` and `summary()`.

# Fitted values

- Once we execute

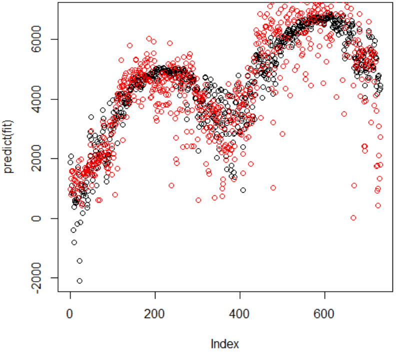
```
fit <- lm(B$cnt ~ B$instant + B$workingday)
```

  the object `fit` contains more than the regression report.

- It contains the **fitted values** $\hat{y}_i$:

```
predict(fit)
plot(predict(fit))
points(B$cnt, col = "red")
```

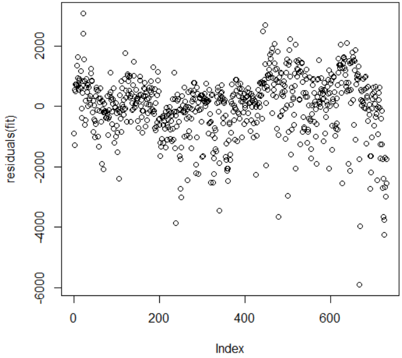
  - `plot()` makes a scatter plot.
  - `points()` add points onto an existing scatter plot.
  - `col = "red"` makes red points.

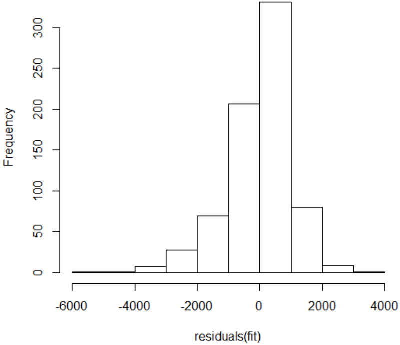


# Residuals

- We may also obtain **residuals**:

```
residuals(fit)
plot(residuals(fit))
hist(residuals(fit))
```



# Road map

- The R programming language.
- Regression in R.
- **Logistic regression**.

# Logistic regression

- So far our regression models always have a **quantitative** variable as the **dependent** variable.
  - Some people call this type of regression **ordinary regression**.
- To have a **qualitative** variable as the dependent variable, ordinary regression does not work.
- One popular remedy is to use **logistic regression**.
  - In general, a logistic regression model allows the dependent variable to have multiple levels.
  - We will only consider **binary variables** in this lecture.
- Let's first illustrate why ordinary regression fails when the dependent variable is binary.

# Example: survival probability

- 45 persons got trapped in a storm during a mountain hiking. Unfortunately, some of them died due to the storm.[4]
- We want to study how the **survival probability** of a person is affected by her/his **gender** and **age**.

| Age | Gender | Survived | Age | Gender | Survived | Age | Gender | Survived |
|---|---|---|---|---|---|---|---|---|
| 23 | Male | No | 23 | Female | Yes | 15 | Male | No |
| 40 | Female | Yes | 28 | Male | Yes | 50 | Female | No |
| 40 | Male | Yes | 15 | Female | Yes | 21 | Female | Yes |
| 30 | Male | No | 47 | Female | No | 25 | Male | No |
| 28 | Male | No | 57 | Male | No | 46 | Male | Yes |
| 40 | Male | No | 20 | Female | Yes | 32 | Female | Yes |
| 45 | Female | No | 18 | Male | Yes | 30 | Male | No |
| 62 | Male | No | 25 | Male | No | 25 | Male | No |
| 65 | Male | No | 60 | Male | No | 25 | Male | No |
| 45 | Female | No | 25 | Male | Yes | 25 | Male | No |
| 25 | Female | No | 20 | Male | Yes | 30 | Male | No |
| 28 | Male | Yes | 32 | Male | Yes | 35 | Male | No |
| 28 | Male | No | 32 | Female | Yes | 23 | Male | Yes |
| 23 | Male | No | 24 | Female | Yes | 24 | Male | No |
| 22 | Female | Yes | 30 | Male | Yes | 25 | Female | Yes |

[4]The data set comes from the textbook *The Statistical Sleuth* by Ramsey and Schafer. The story has been modified.

# Descriptive statistics

- Overall survival probability is $\frac{20}{45} = 44.4\%$.
- Survival or not seems to be affected by gender.

| Group | Survivals | Group size | Survival probability |
|---|---|---|---|
| Male | 10 | 30 | 33.3% |
| Female | 10 | 15 | 66.7% |

- Survival or not seems to be affected by age.

| Age class | Survivals | Group size | Survival probability |
|---|---|---|---|
| $[10, 20)$ | 2 | 3 | 66.7% |
| $[21, 30)$ | 11 | 22 | 50.0% |
| $[31, 40)$ | 4 | 8 | 50.0% |
| $[41, 50)$ | 3 | 7 | 42.9% |
| $[51, 60)$ | 0 | 2 | 0.0% |
| $[61, 70)$ | 0 | 3 | 0.0% |

- May we do better? May we predict one's survival probability?

# Ordinary regression is problematic

- Immediately we may want to construct a linear regression model

$$survival_i = \beta_0 + \beta_1 age_i + \beta_2 female_i + \epsilon_i.$$

  where *age* is one's age, *gender* is 0 if the person is a male or 1 if female, and *survival* is 1 if the person is survived or 0 if dead.
- By running

```
d <- read.table("survival.txt", header = TRUE)
fitWrong <- lm(d$survival ~ d$age + d$female)
summary(fitWrong)
```

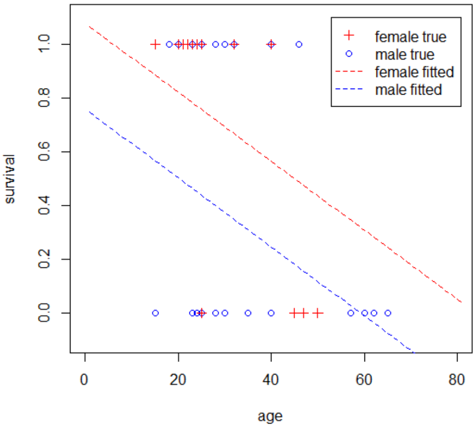
  we may obtain the regression line

$$survival = 0.746 - 0.013 age + 0.319 female.$$

  Though $R^2 = 0.1642$ is low, both variables are significant.

# Ordinary regression is problematic

- The regression model gives us "predicted survival probability."
  - For a man at 80, the "probability" becomes $0.746 - 0.013 \times 80 = -0.294$, which is **unrealistic**.
- In general, it is very easy for an ordinary regression model to generate predicted "probability" not within 0 and 1.

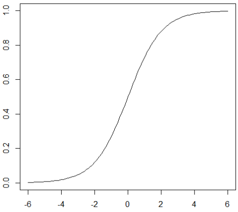


# Logistic regression

- The right way to do is to do **logistic regression**.
- Consider the age-survival example.
  - We still believe that the smaller age increases the survival probability.
  - However, not in a linear way.
  - It should be that when one is **young enough**, being younger does not help too much.
  - The **marginal benefit** of being younger should be decreasing.
  - The **marginal loss** of being older should also be decreasing.
- One particular functional form that exhibits this property is

$$y = \frac{e^x}{1+e^x} \quad \Leftrightarrow \quad \log\left(\frac{y}{1-y}\right) = x$$

  - $x$ can be anything in $(-\infty, \infty)$.
  - $y$ is limited in $[0, 1]$.



# Logistic regression

- We **hypothesize** that independent variables $x_i$s affect $\pi$, the probability for $y$ to be 1, in the following form:[5]

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p.$$

- The equation looks scaring. Fortunately, R is powerful.
- In R, all we need to do is to switch from `lm()` to `glm()` with an additional argument `binomial`.
  - `lm` is the abbreviation of "linear model."
  - `glm()` is the abbreviation of "generalized linear model."

[5] The logistic regression model searches for coefficients to make the curve fit the given data points in the best way. The details are far beyond the scope of this course.

# Logistic regression in R

- By executing

  ```
  fitRight <- glm(d$survival ~ d$age + d$female, binomial)
  summary(fitRight)
  ```

  we obtain the regression report.
- Some information is new, but the following is familiar:

  ```
  Coefficients:
              Estimate Std. Error z value Pr(>|z|)
  (Intercept)  1.63312    1.11018   1.471   0.1413
  d$age       -0.07820    0.03728  -2.097   0.0359 *
  d$female     1.59729    0.75547   2.114   0.0345 *
  ---
  Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
  ```
- Both variables are **significant**.

# The Logistic regression curve

- The estimated curve is

$$\log\left(\frac{\pi}{1-\pi}\right) = 1.633 - 0.078 age + 1.597 female,$$

or equivalently,

$$\pi = \frac{\exp(1.633 - 0.078 age + 1.597 female)}{1 + \exp(1.633 - 0.078 age + 1.597 female)},$$

where $\exp(z)$ means $e^z$ for all $z \in \mathbb{R}$.

# The Logistic regression curve

- The curves can be used to do **prediction**.
- For a man at 80, $\pi$ is

$$\frac{\exp(1.633-0.078\times 80)}{1+\exp(1.633-0.078\times 80)},$$

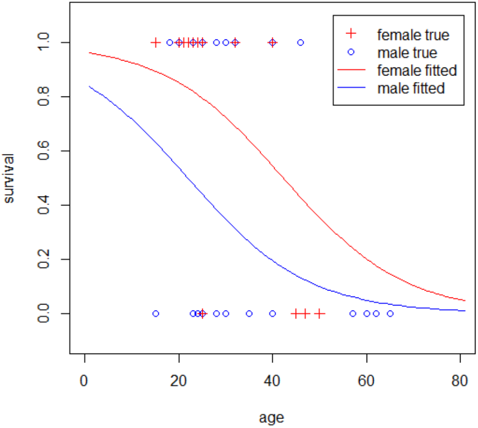
which is 0.0097.

- For a woman at 60, $\pi$ is

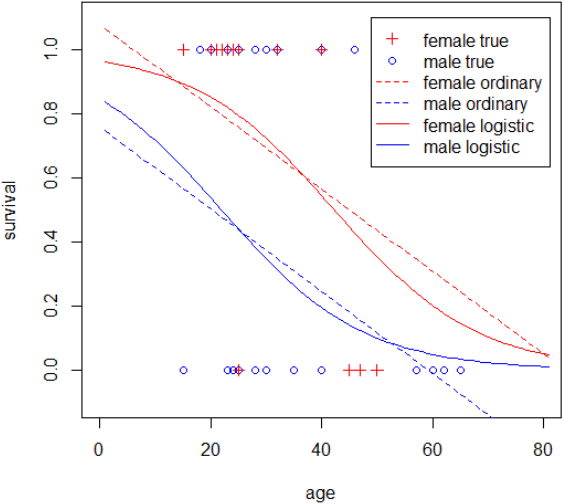
$$\frac{\exp(1.633-0.078\times 60+1.597)}{1+\exp(1.633-0.078\times 60+1.597)},$$

which is 0.1882.

- $\pi$ is always in $[0,1]$. There is no problem for interpreting $\pi$ as a probability.



# Comparisons



# Interpretations

- The estimated curve is

$$\log\left(\frac{\pi}{1-\pi}\right) = 1.633 - 0.078 age + 1.597 female.$$

  Any implication?
  - $-0.078 age$: Younger people will survive more likely.
  - $1.597 female$: Women will survive more likely.
- In general:
  - Use the ***p*-values** to determine the significance of variables.
  - Use the **signs** of coefficients to give qualitative implications.
  - Use the **formula** to make predictions.

# Model selection

- Recall that in ordinary regression, we use $R^2$ and adjusted $R^2$ to assess the usefulness of a model.
- In logistic regression, we do not have $R^2$ and adjusted $R^2$.
- We have **deviance** instead.
  - In a regression report, the **null deviance** can be considered as the total estimation errors without using any independent variable.
  - The **residual deviance** can be considered as the total estimation errors by using the selected independent variables.
  - Ideally, the residual deviance should be **small**.[6]

[6] To be more rigorous, the residual deviance should also be close to its degree of freedom. This is beyond the scope of this course.

# Deviances in the regression report

- The null and residual deviances are provided in the regression report.
- For `glm(d$survival ~ d$age + d$female, binomial)`, we have

```
    Null deviance: 61.827  on 44  degrees of freedom
Residual deviance: 51.256  on 42  degrees of freedom
```

- Let's try some models:

| Independent variable(s) | Null deviance | Residual deviance |
|---|---|---|
| *age* | 61.827 | 56.291 |
| *female* | 61.827 | 57.286 |
| *age*, *female* | 61.827 | 51.256 |
| *age*, *female*, $age \times female$ | 61.827 | 47.346 |

  - Using *age* only is better than using *female* only.
- How to compare models with different numbers of variables?

# Deviances in the regression report

- Adding variables will **always reduce** the residual deviance.
- To take the number of variables into consideration, we may use **Akaike Information Criterion** (AIC).
- AIC is also included in the regression report:

| Independent variable(s) | Null deviance | Residual deviance | AIC |
|---|---|---|---|
| *age* | 61.827 | 56.291 | 60.291 |
| *female* | 61.827 | 57.286 | 61.291 |
| *age*, *female* | 61.827 | 51.256 | 57.256 |
| *age*, *female*, $age \times female$ | 61.827 | 47.346 | 55.346 |

- AIC is only used to compare **nested** models.
  - Two models are nested if one's variables are form a subset of the other's.
  - Model 4 is better than model 3 (based on their AICs).
  - Model 3 is better than either model 1 or model 2 (based on their AICs).
  - Model 1 and 2 cannot be compared (based on their AICs).