13

데이터분석방법론(1)

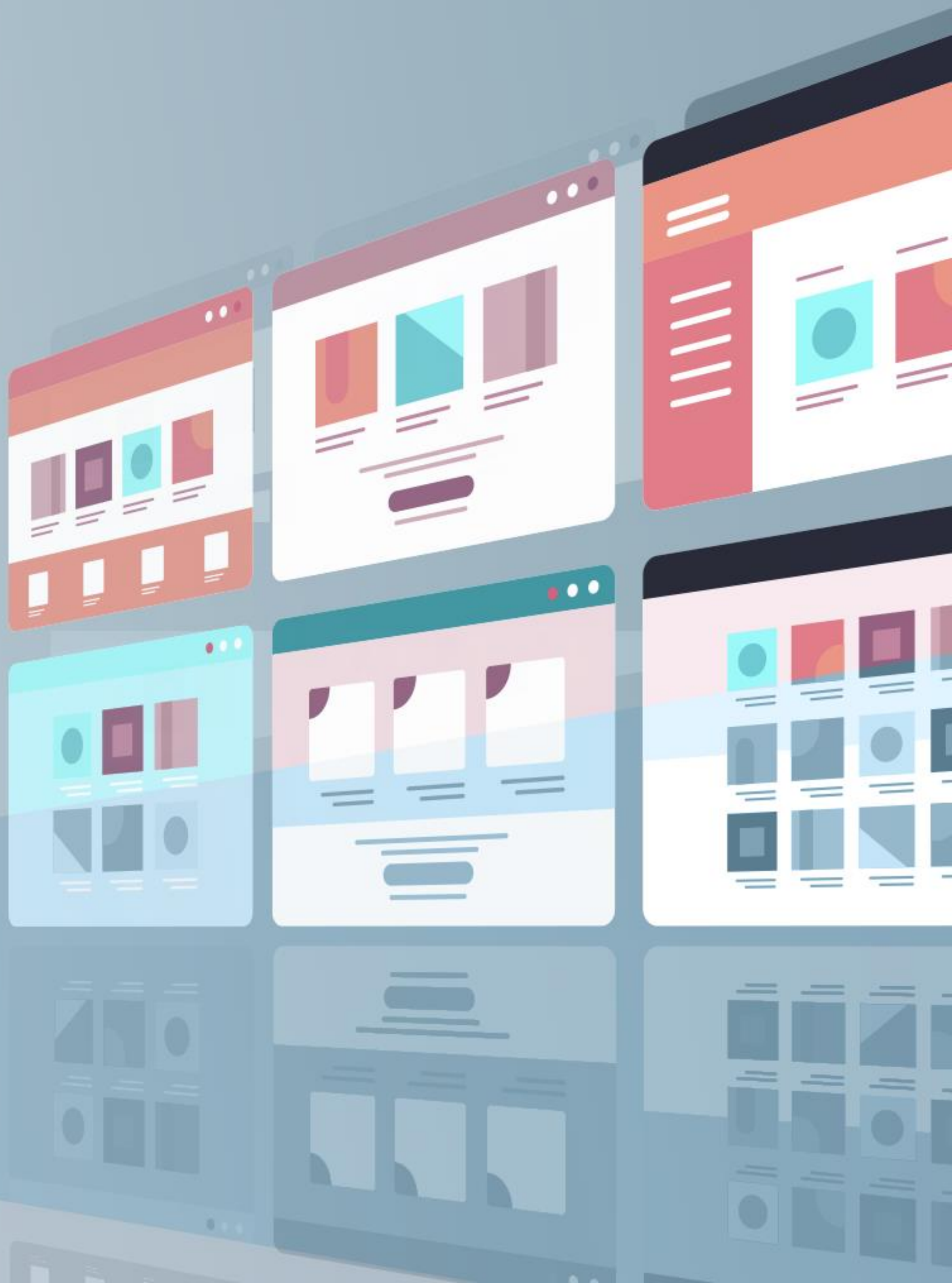
Logistic Regression

통계데이터과학과 장영재 교수

학습목차

01

The Model



1. Logistic Regression

- Sometimes you wish to model binary outcomes, variables that can have only two possible values such as diseased or non-diseased, success or failure ,and so forth.

- It is not really attractive to use additive models for probabilities since they have a limited range and regression models could predict off-scale values below zero or above 1.

- It makes better sense to model the probabilities on a transformed scale; this is what is done in logistic regression analysis.

2. Generalized linear model

- Logistic regression analysis belongs to the class of generalized linear models.

- These models are characterized by their response distribution (here the binomial distribution) and a link function.

In a logistic regression analysis, the link function is

$$logit(p) = \log(\frac{p}{1-p})$$

Logistic regression model

$$\log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

2. Generalized linear model

◆ Data description

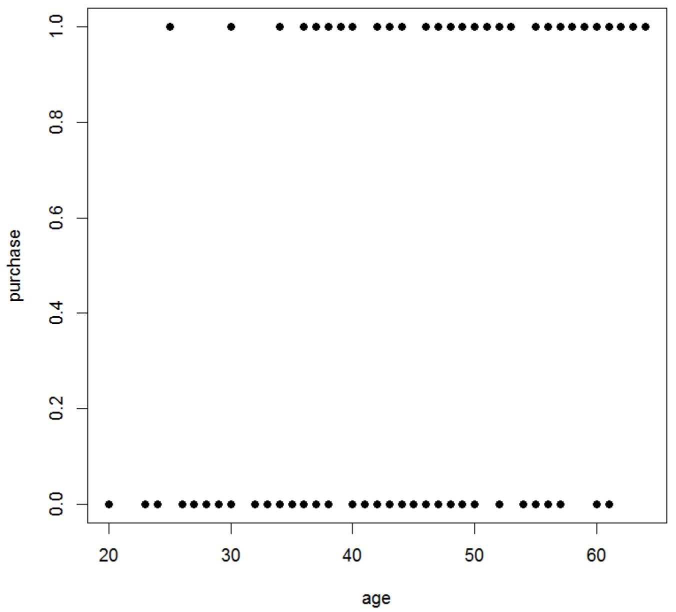
- Response variable 'purchase' : 0 = No purchase, 1 = purchase
- The total number of observations is 100.

	A	B	C
1	id	age	purchase
2	1	20	0
3	2	23	0
4	3	24	0
5	4	25	1
6	5	26	0
7	6	27	0
8	7	27	0
9	8	28	0
10	9	29	0
11	10	29	0
12	11	30	0
13	12	30	0
14	13	30	0
15	14	30	1
16	15	32	0

2. Generalized linear model

(1) Scatter plot

```
> library(xlsx)
> drug.data = read.xlsx("c:/data/mva/drug.xlsx", 1)
> head(drug.data)
  id age purchase
1  1  20        0
2  2  23        0
3  3  24        0
4  4  25        1
5  5  26        0
6  6  27        0
> attach(drug.data)
> plot(age, purchase, pch=19)
```



The higher the age, the more it tends to take the value of y = 1, but it is difficult to clearly state the relationship between the two variables age and purchase.

2. Generalized linear model

(2) Grouping ages

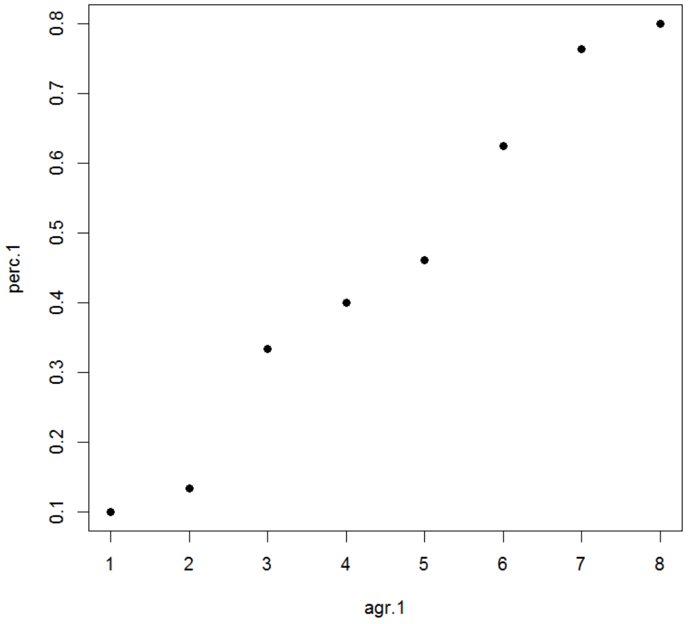
```
> #Recoding
> agr = age
> agr[agr >= 20 & agr <= 29 ] = 1
> agr[agr >= 30 & agr <= 34 ] = 2
> agr[agr >= 35 & agr <= 39 ] = 3
> agr[agr >= 40 & agr <= 44 ] = 4
> agr[agr >= 45 & agr <= 49 ] = 5
> agr[agr >= 50 & agr <= 54 ] = 6
> agr[agr >= 55 & agr <= 59 ] = 7
> agr[agr >= 60 & agr <= 64 ] = 8
```

```
> purchase.table = table(agr, purchase)
> purchase.table
   purchase
agr 0   1
  1  9   1
  2 13   2
  3  8   4
  4  9   6
  5  7   6
  6  3   5
  7  4  13
  8  2   8
```

2. Generalized linear model

(3) Grouped variable agr and variable purchas

```
> percent.table = prop.table(purchase.table, 1)
> percent.table
   purchase
agr         0         1
  1 0.9000000 0.1000000
  2 0.8666667 0.1333333
  3 0.6666667 0.3333333
  4 0.6000000 0.4000000
  5 0.5384615 0.4615385
  6 0.3750000 0.6250000
  7 0.2352941 0.7647059
  8 0.2000000 0.8000000
> perc.1 = percent.table[,2]
> agr.1 = rownames(percent.table)
> agr.1 = as.numeric(agr.1)
> plot(agr.1, perc.1, pch=19)
```

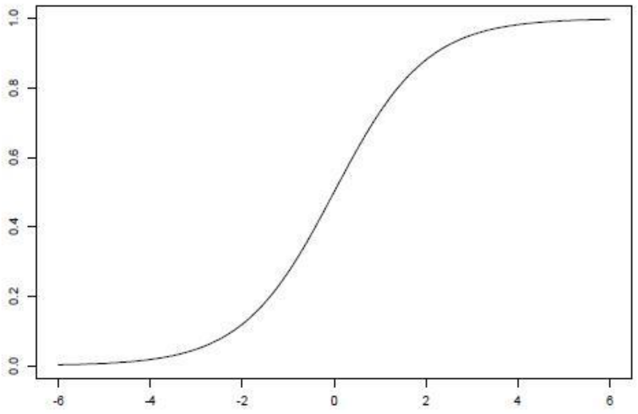


- Probability of purchase increases as the age gets bigger with S-shape.

- Logistic function can be applied to a S-shape function.

2. Generalized linear model

◆ Logistic function



- Dependent variable Y : binary (0 or 1)
  and one independent variable X
- $P(Y=1|X)$: Probability of Y=1 given X
- Logistic function : S-shape curve function which converges to 1 as X increases, converges to as X decreases

$$P(Y=1|X)=\frac{\exp(\beta_0+\beta_1 X)}{1+\exp(\beta_0+\beta_1 X)}$$

2. Generalized linear model

- Transformation of logistic function

  Let $P(Y=1|X)=p$

$$\ln\left(\frac{p}{1-p}\right)=\beta_0+\beta_1 X$$

  Here, $\frac{p}{1-p}$ is called odds

$$\text{odds} = \frac{p}{1-p} \rightarrow p=\frac{odds}{1+odds}$$

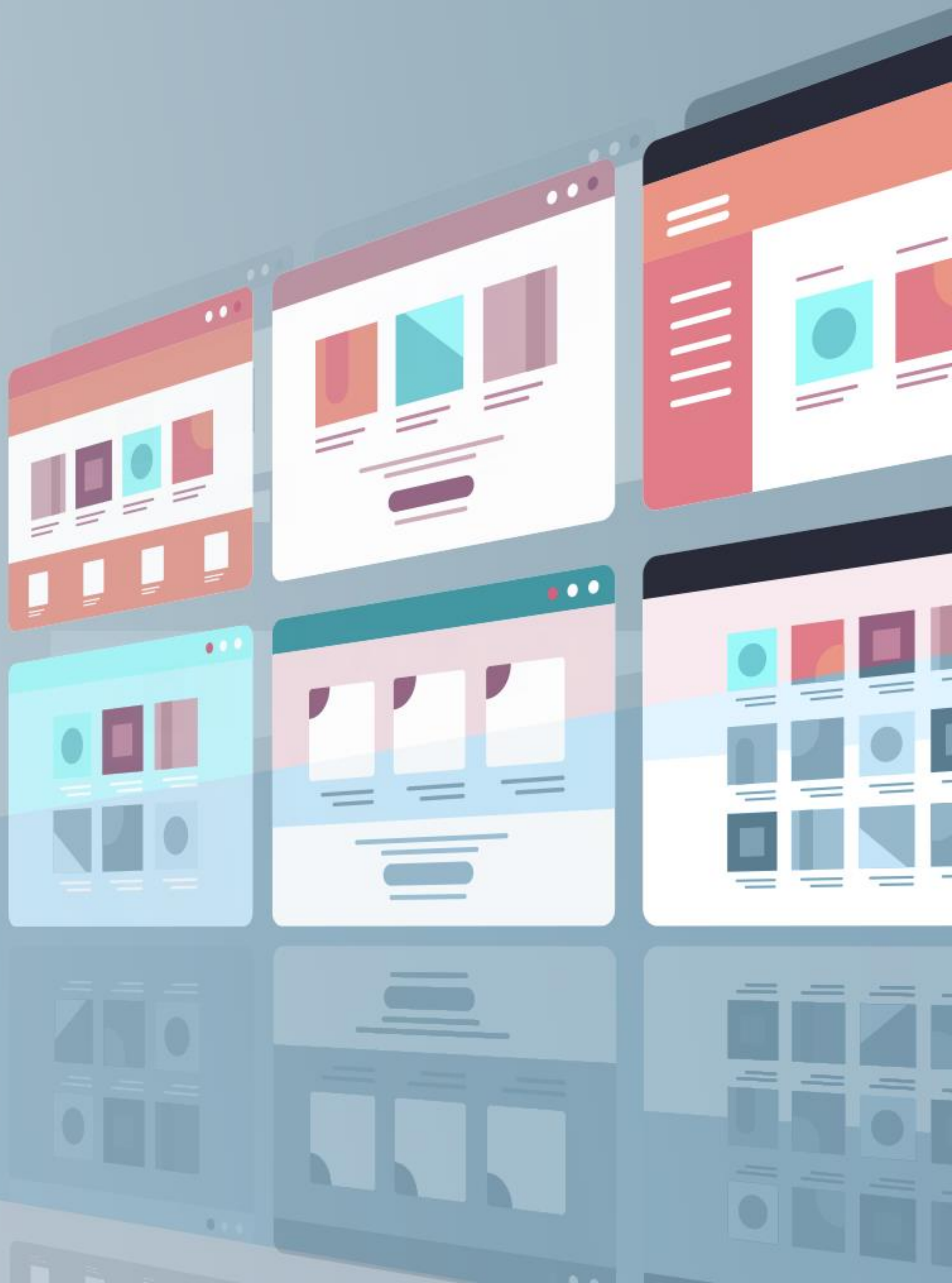
- Meaning of odds

  Ex) In a sports game, let odds of team A beating team B is 4, then team A has a four times higher chance(Probability) of beating team B.

$$p=\frac{4}{1+4}=0.8, \quad \text{namely, odds}=\frac{0.8}{1-0.8}=4$$

02

Example of Logistic Regression



5. Logistic regression using raw data

- Descriptive statistics of Juul's data (included in the "ISwR" package)

```
> summary(juul)
      age            menarche          sex             igf1          tanner
 Min.   : 0.170   Min.   :1.000   Min.   :1.000   Min.   : 25.0   I   :515
 1st Qu.: 9.053   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:202.2   II  :103
 Median :12.560   Median :1.000   Median :2.000   Median :313.5   III : 72
 Mean   :15.095   Mean   :1.476   Mean   :1.534   Mean   :340.2   IV  : 81
 3rd Qu.:16.855   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:462.8   V   :328
 Max.   :83.000   Max.   :2.000   Max.   :2.000   Max.   :915.0   NA's:240
 NA's   :5        NA's   :635     NA's   :5       NA's   :321
    testvol
 Min.   : 1.000
 1st Qu.: 1.000
 Median : 3.000
 Mean   : 7.896
 3rd Qu.:15.000
 Max.   :30.000
 NA's   :859
```

```
> juul$menarche <- factor(juul$menarche, labels=c("No","Yes"))
> juul$tanner <- factor(juul$tanner)
```

```
> juul.girl <- subset(juul,age>8 & age<20 &
+ complete.cases(menarche))
```

: a subset of data consisting of 8-20-year-old girls.

5. Logistic regression using raw data

- Analyze menarche as a function of age

```
> summary(glm(menarche~age,binomial, data=juul.girl))
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.32759  -0.18998   0.01253   0.12132   2.45922

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -20.0132     2.0284  -9.867   <2e-16 ***
age           1.5173     0.1544   9.829   <2e-16 ***
---

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 719.39  on 518  degrees of freedom
Residual deviance: 200.66  on 517  degrees of freedom
AIC: 204.66
```

: estimate the median menarcheal age as the age where logit p = 0.

$-20.0132 + 1.5173 \times \text{age} = 0$

=> $20.0132/1.5173 = 13.19$ years

5. Logistic regression using raw data

- A more complicated analysis is obtained by including the Tanner stage of puberty in the model.

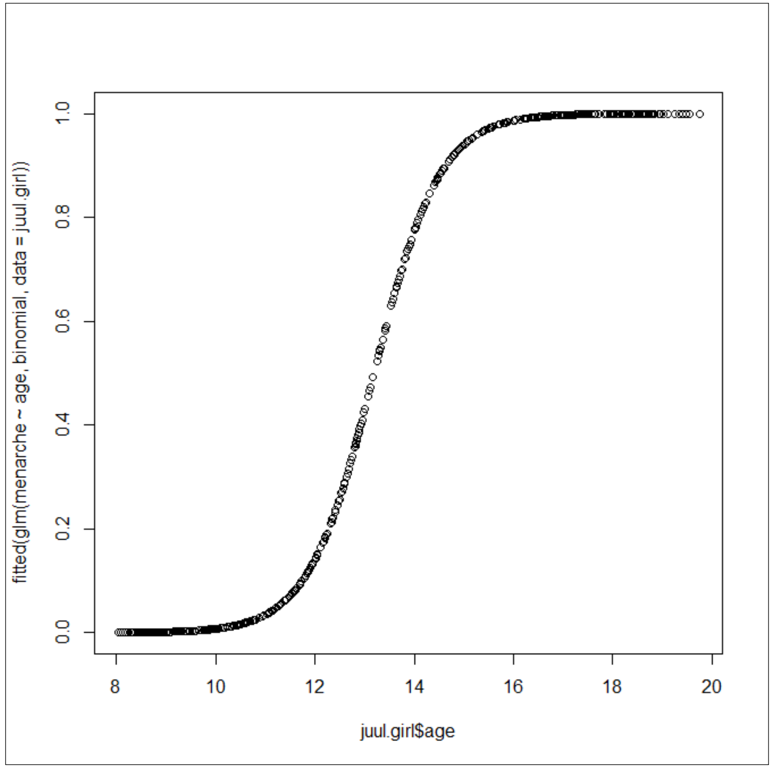
```
> summary(glm(menarche~age+tanner,binomial, data=juul.girl))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -13.7758     2.7630  -4.986 6.17e-07 ***
age           0.8603     0.2311   3.723 0.000197 ***
tannerII     -0.5211     1.4846  -0.351 0.725609
tannerIII     0.8264     1.2377   0.668 0.504313
tannerIV      2.5645     1.2172   2.107 0.035132 *
tannerV       5.1897     1.4140   3.670 0.000242 ***
```

There are a couple of significant z-values, so you would expect that the tanner variable has some effect. The formal test, however, must be obtained from the deviances:

6. Prediction

- In the analysis of menarche, the primary interest is probably in seeing a plot of the expected probabilities versus age

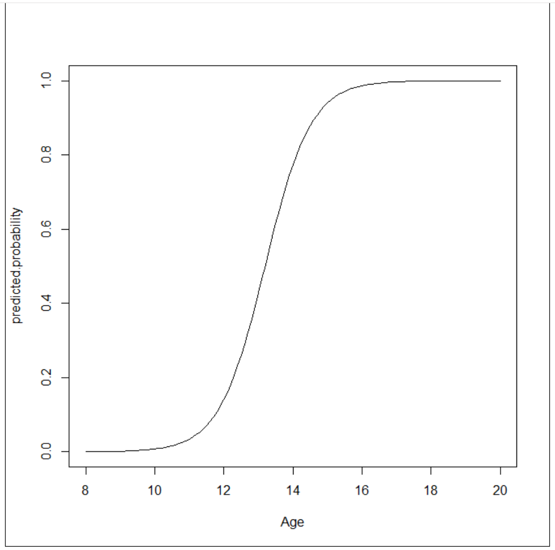
```
> head(juul.girl)
      age menarche sex igf1 tanner testvol
167  8.96       No   2   NA      I      NA
343 13.01       No   2  682     II      NA
743  8.03       No   2   NA      I      NA
744  8.08       No   2   NA      I      NA
745  8.13       No   2  210      I      NA
746  8.17       No   2  564   <NA>      NA
> plot(juul.girl$age,
       fitted(glm(menarche~age,binomial,
              data=juul.girl)))
```



6. Prediction

- A more ambitious plan

```
> glm.menarche <- glm(menarche~age, binomial, data=juul.girl)
> Age <- seq(8,20,.1)
> newages <- data.frame(age=Age)
> predicted.prob <- predict(glm.menarche,
+                           newages,type="resp")
> head(predicted.prob)
           1            2            3            4            5            6 
0.0003800216 0.0004422586 0.0005146830 0.0005989606 0.0006970286 0.0008111404 
> plot(predicted.prob ~ Age, type="l")
```



7. Model checking

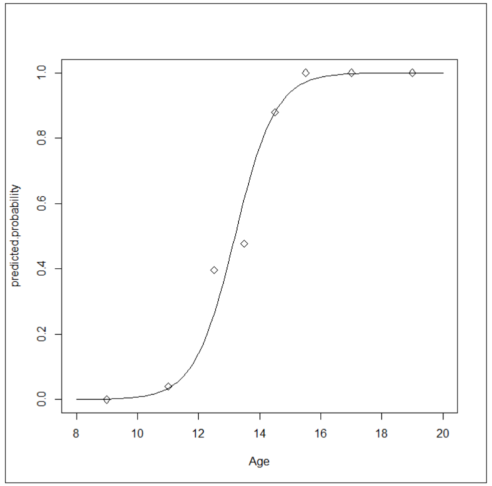
- For complex models with continuous background variables, it becomes more difficult to perform an adequate model check.
- For this case, you might try subdividing the x-axis in a number of intervals and see how the counts in each interval fit with the expected probabilities.

```
> age.group <- cut(juul.girl$age,c(8,10,12,13,14,15,16,18,20))
> tb <- table(age.group, juul.girl$menarche)
> tb
age.group  No Yes
  (8,10]  100   0
  (10,12]  97   4
  (12,13]  32  21
  (13,14]  22  20
  (14,15]   5  36
  (15,16]   0  31
  (16,18]   0 105
  (18,20]   0  46
```

7. Model checking

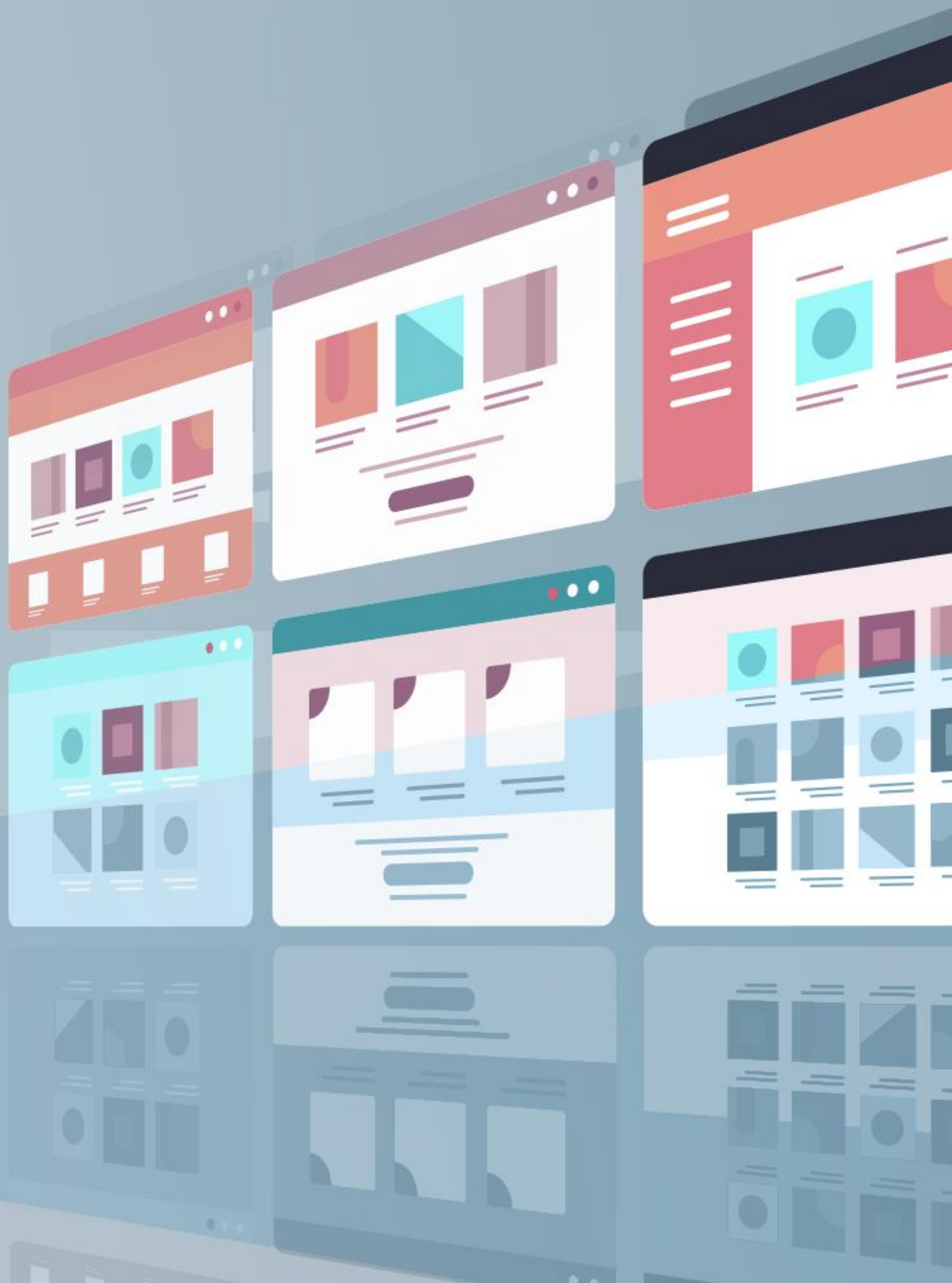
- For complex models with continuous background variables, it becomes more difficult to perform an adequate model check.
- For this case, you might try subdividing the x-axis in a number of intervals and see how the counts in each interval fit with the expected probabilities.

```
> rel.freq <- prop.table(tb,1)[,2]
> rel.freq
    (8,10]    (10,12]    (12,13]    (13,14]    (14,15]    (15,16]    (16,18]   (18,20]
0.00000000 0.03960396 0.39622642 0.47619048 0.87804878 1.00000000 1.00000000 1.0000000
> points(rel.freq ~ c(9,11,12.5,13.5,14.5,15.5,17,19),pch=5)
```



03

Logistic Regression on Tabular Data



1. The tabular data

- Example of hypertension (Altman, 1991)

```
> no.yes <- c("No","Yes")
> smoking <- gl(2,1,8,no.yes)
> obesity <- gl(2,2,8,no.yes)
> snoring <- gl(2,4,8,no.yes)
> n.tot <- c(60,17,8,2,187,85,51,23)
> n.hyp <- c(5,2,1,0,35,13,15,8)
> data.frame(smoking,obesity,snoring,n.tot,n.hyp)
  smoking obesity snoring n.tot n.hyp
1      No      No      No    60     5
2     Yes      No      No    17     2
3      No     Yes      No     8     1
4     Yes     Yes      No     2     0
5      No      No     Yes   187    35
6     Yes      No     Yes    85    13
7      No     Yes     Yes    51    15
8     Yes     Yes     Yes    23     8
```

2. Two ways for logistic regression

- R is able to fit logistic regression analyses for tabular data in two different ways.

```
> hyp.tbl <- cbind(n.hyp,n.tot-n.hyp)
> hyp.tbl
     n.hyp
[1,]     5  55
[2,]     2  15
[3,]     1   7
[4,]     0   2
[5,]    35 152
[6,]    13  72
[7,]    15  36
[8,]     8  15
```

3. Method 1 : Original data

<Method 1>

You have to specify the response as a matrix, where one column is the number of "diseased" and the other is the number of "healthy" (or "success" and "failure", depending on context)

```
> hyp.tbl <- cbind(n.hyp,n.tot-n.hyp)
> hyp.tbl
     n.hyp
[1,]     5  55
[2,]     2  15
[3,]     1   7
[4,]     0   2
[5,]    35 152
[6,]    13  72
[7,]    15  36
[8,]     8  15
```

3. Method 1 : Results

<Results>

```
> glm(hyp.tbl~smoking+obesity+snoring,family=binomial("logit"))

Call:  glm(formula = hyp.tbl ~ smoking + obesity + snoring, family
=binomial("logit"))

Coefficients:
(Intercept)   smokingYes   obesityYes   snoringYes
   -2.37766     -0.06777      0.69531      0.87194

Degrees of Freedom: 7 Total (i.e. Null);  4 Residual
Null Deviance:      14.13
Residual Deviance: 1.618        AIC: 34.54
```

4. Method 2 : Proportion as the response variable

<Method 2>

The other way to specify a logistic regression model is to give the proportion of diseased in each cell:

```
> prop.hyp <- n.hyp/n.tot
> glm.hyp <- glm(prop.hyp~smoking+obesity+snoring,  binomial,weights=n.tot)
> summary(glm.hyp)
...
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.37766    0.38018  -6.254    4e-10 ***
smokingYes  -0.06777    0.27812  -0.244   0.8075
obesityYes   0.69531    0.28509   2.439   0.0147 *
snoringYes   0.87194    0.39757   2.193   0.0283 *
---

    Null deviance: 14.1259  on 7  degrees of freedom
Residual deviance:  1.6184  on 4  degrees of freedom
AIC: 34.537

Number of Fisher Scoring iterations: 4
```

4. Method 2 : Results

<Results>

```
Deviance Residuals:
       1         2         3         4         5         6         7
-0.04344   0.54145  -0.25476  -0.80051   0.19759  -0.46602  -0.21262
       8
 0.56231
```

; This is the contribution of each cell of the table to the deviance of the model (the deviance corresponds to the sum of squares in linear normal models), with a sign according to whether the observation is larger or smaller than expected.

They can be used to pinpoint cells that are particularly poorly fitted, but you have to be wary of the interpretation in sparse tables.

4. Method 2 : Results

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.37766    0.38018  -6.254    4e-10 ***
smokingYes  -0.06777    0.27812  -0.244   0.8075
obesityYes   0.69531    0.28509   2.439   0.0147 *
snoringYes   0.87194    0.39757   2.193   0.0283 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

; This is the table of primary interest. Here, we get estimates of the regression coefficients, standard errors of same, and tests for whether each regression coefficient can be assumed to be zero.

$$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -2.37766 - 0.06777\,smokingY + 0.69531\,obeY + 0.87194\,snorY$$

4. Method 2 : Results

```
Null deviance: 14.1259  on 7  degrees of freedom
Residual deviance:  1.6184  on 4  degrees of freedom
AIC: 34.537
```

Null deviance : Constant only model (difference between estimates and observations)

Residual deviance : Independent variables included

H_0 : Model is correct.

H_1 : Model is not correct.

p-value =0.805, accept H_0.

$$p-value = P(\chi^2 > 1.6184)$$
$$= 1 - P(\chi^2 \le 1.6184)$$

```
> 1-pchisq(1.6184, 4)
[1] 0.8054813
```

The 5% significance limit : 9.49

```
> qchisq(0.95,4)
[1] 9.487729
```

4. Method 2 : Results

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.37766    0.38018  -6.254    4e-10 ***
smokingYes  -0.06777    0.27812  -0.244   0.8075
obesityYes   0.69531    0.28509   2.439   0.0147 *
snoringYes   0.87194    0.39757   2.193   0.0283 *
---
```

The z test in the table of regression coefficients immediately shows that the model can be simplified by removing smoking.

```
> glm.hyp2 <- glm(prop.hyp~obesity+snoring,family=binomial, weights=n.tot)
> summary(glm.hyp2)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.3921     0.3757  -6.366 1.94e-10 ***
obesityYes    0.6954     0.2851   2.440   0.0147 *
snoringYes    0.8655     0.3967   2.182   0.0291 *
---
Residual deviance:  1.6781  on 5  degrees of freedom
```

```
> 1-pchisq(1.6781, 5)
[1] 0.8916472
```

5. Analysis of deviance table

- The analysis of deviance table corresponds to ANOVA tables.

Deviance tables correspond to ANOVA tables for multiple regression analyses and are generated like these with the anova function:

```
> glm.hyp <- glm(hyp.tbl~smoking+obesity+snoring,binomial)
> anova(glm.hyp, test="Chisq")
        Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                        7    14.1259
smoking  1   0.0022         6    14.1237 0.962724
obesity  1   6.8274         5     7.2963 0.008977 **
snoring  1   5.6779         4     1.6184 0.017179 *
```

The Deviance column gives differences between models as variables are added to the model in turn. The deviances are approximately χ^2-distributed with the stated degrees of freedom.

Since the snoring variable on the last line is significant, it may not be removed from the model and we cannot use the table to justify model reductions.

5. Analysis of deviance table

If, however, the terms are rearranged so that smoking comes last, we get a deviance-based test for removal of that variable:

```
> glm.hyp <- glm(hyp.tbl~snoring+obesity+smoking,binomial)
> anova(glm.hyp, test="Chisq")
        Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                        7    14.1259
snoring  1   6.7887         6     7.3372 0.009174 **
obesity  1   5.6591         5     1.6781 0.017365 *
smoking  1   0.0597         4     1.6184 0.806938
```

From this you can read that smoking is removable, whereas obesity is not, after removal of smoking.

The information in the deviance tables is fundamentally the same as that given by the z tests in the table of regression coefficients.

6. Presentation as odds-ratio estimates

- In parts of the epidemiological literature, it has become traditional to present logistic regression analyses in terms of odds ratios. In the case of a quantitative covariate, this means odds ratio per unit change in the covariate.
- Since standard errors make little sense after the transformation, it is also customary to give confidence intervals instead.

```
> exp(cbind(OR=coef(glm.hyp), confint(glm.hyp)))
Waiting for profiling to be done...
                    OR      2.5 %   97.5 %
(Intercept) 0.09276726 0.04063914 0.183823
snoringYes  2.39154432 1.15660384 5.605594
obesityYes  2.00432951 1.13345994 3.478922
smokingYes  0.93447081 0.53379700 1.594628
```

7. Prediction

- The predict function works for generalized linear models, too.

```
> prop.hyp
[1] 0.08333333 0.11764706 0.12500000 0.00000000 0.18716578 0.15294118
[7] 0.29411765 0.34782609
> glm.hyp2 <- glm(prop.hyp~obesity+snoring,family=binomial, weights=n.tot)
> predict(glm.hyp2)   # these numbers are on the logit scale
         1          2          3          4          5          6          7
-2.3920763 -2.3920763 -1.6966575 -1.6966575 -1.5266180 -1.5266180 -.8311991
         8
-0.8311991
# predicted values on the response (probabilities)
> predict(glm.hyp, type="response")
         1          2          3          4          5          6          7
0.08489206 0.07977292 0.15678429 0.14803121 0.18157364 0.17171843 0.30780259
         8
0.29355353
```

8. Model checking

- For tabular data it is obvious to try to compare observed and fitted proportions.

```
> fitted(glm.hyp)
         1          2          3          4          5          6          7          8 
0.08489206 0.07977292 0.15678429 0.14803121 0.18157364 0.17171843 0.30780259 0.29355353 
> prop.hyp
[1] 0.08333333 0.11764706 0.12500000 0.00000000 0.18716578 0.15294118 0.29411765 0.34782609
```

- The problem with this is that you get no feeling for how well the relative frequencies are determined. It can be better to look at observed and expected counts instead.

```
> data.frame(fit=fitted(glm.hyp)*n.tot,n.hyp,n.tot)
        fit n.hyp n.tot
1 5.0935236     5    60
2 1.3561397     2    17
3 1.2542744     1     8
4 0.2960624     0     2
...
```

다음시간 안내

14

Quantile Regression