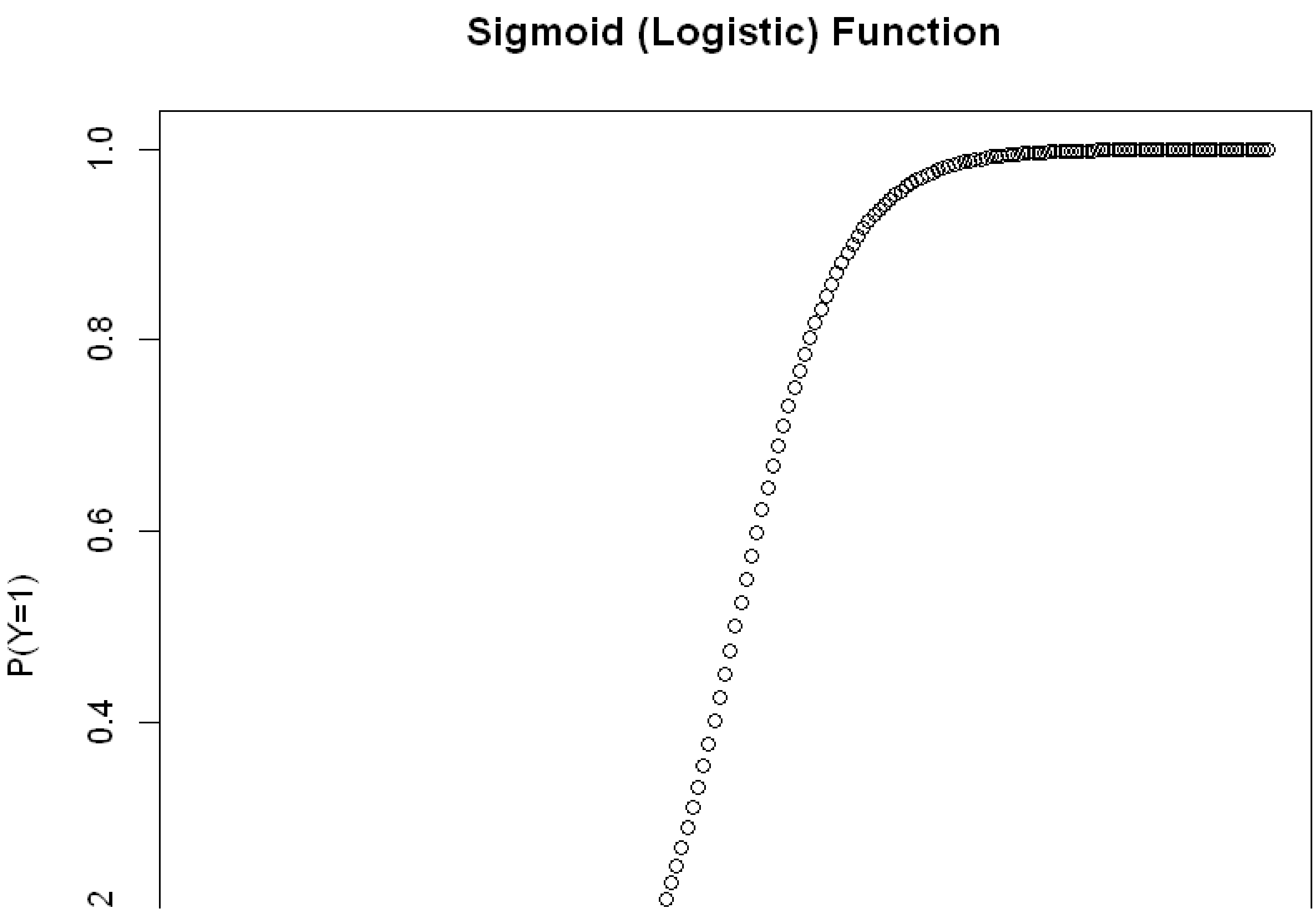
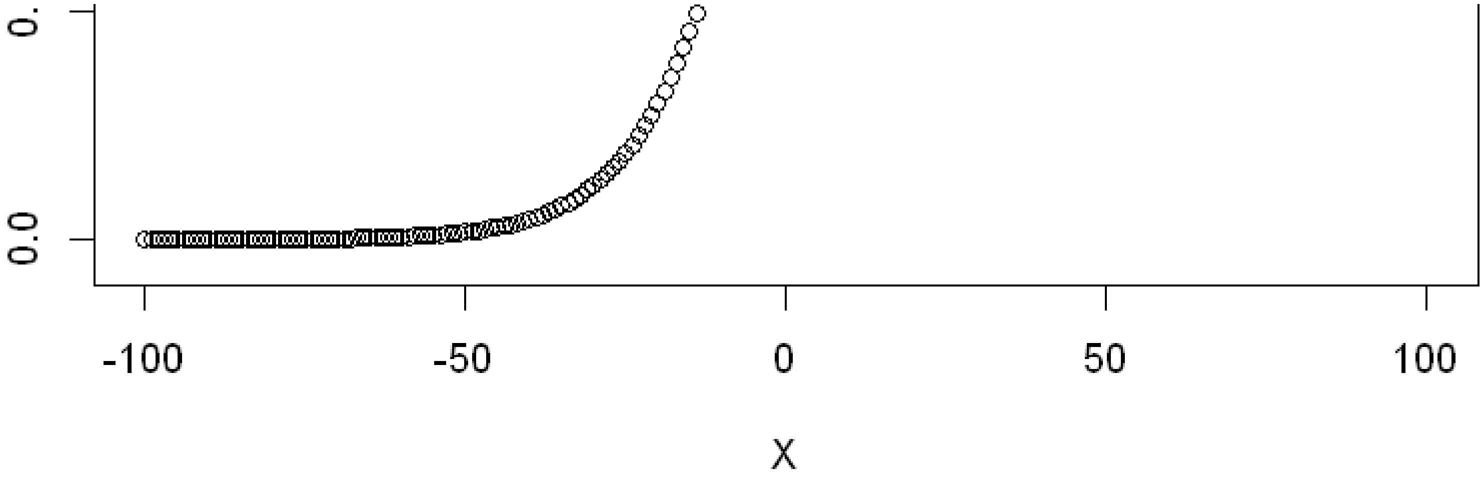
로지스틱 회귀 분석 실습

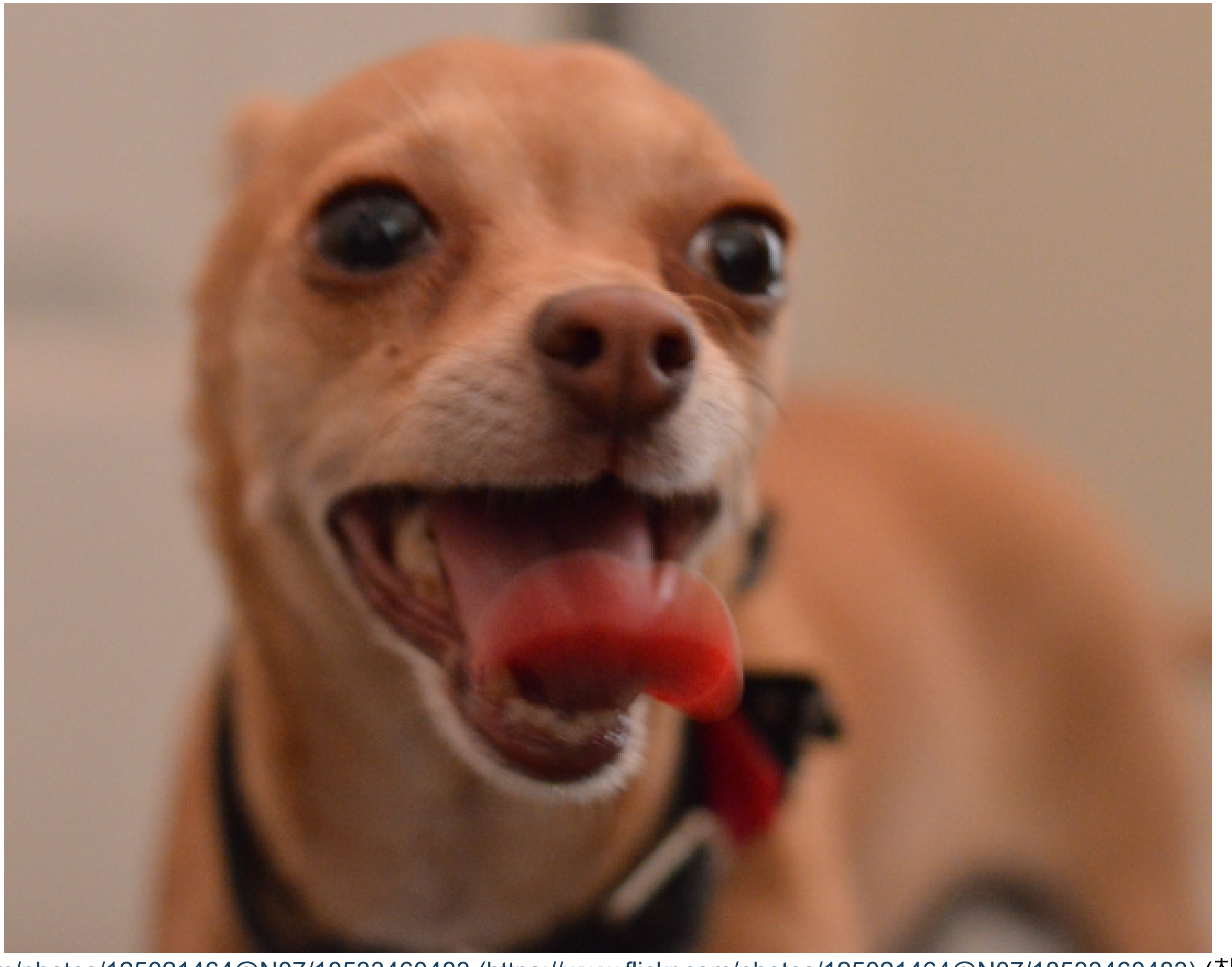
In [6]:
```
x <- c(-100:100)
b = 0  # intercept(절편)
m = 0.1  # slope
y = exp((b + m*x)) / (1 + exp((b+m*x)))
plot(x, y, xlab="X", ylab="P(Y=1)")
title(main="Sigmoid (Logistic) Function")
```





우리집 강아지 헐떡거림 - 로지스틱 회귀 모형

```
여러분의 개가 헐떡거릴때가 있다. 어느 경우에 헐떡일까? 조사해 보니, 온도에 의해 영향을 많이 받는 것 같았다.
```



https://www.flickr.com/photos/125021464@N07/18533460483 (https://www.flickr.com/photos/125021464@N07/18533460483) (참조)

In [7]:
```
temp <- c(10,13.2,12.1,15,17,18,15,14.9,16.0,19,
          21,22,24.5,28,27,20.1,25.6,27.2,29,28.2)
dog.panting<-c(0,0,0,0,0,1,0,1,0,0,
               1,1,1,1,0,1,0,1,1,1)
```

In [8]:
```
data <- data.frame(temp, dog.panting)
data
```

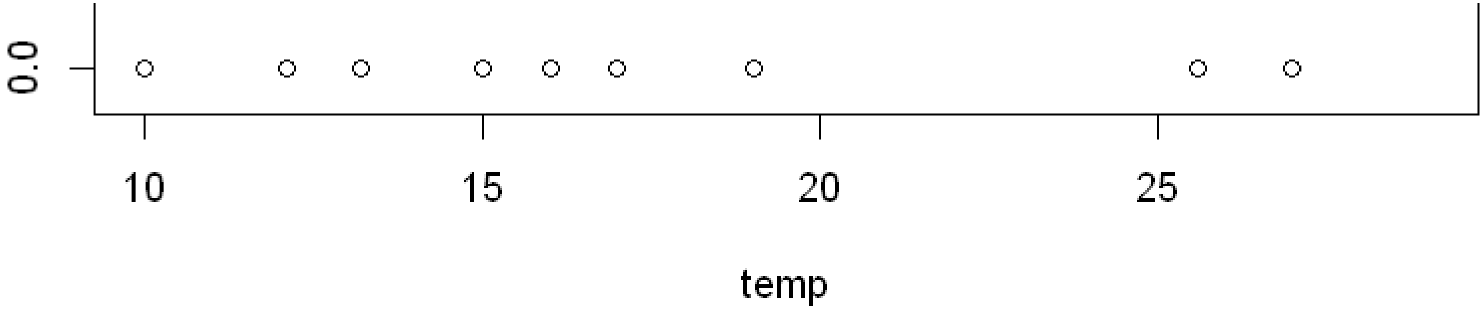
temp	dog.panting
10.0	0
13.2	0
12.1	0
15.0	0
17.0	0
18.0	1
15.0	0
14.9	1
16.0	0
19.0	0
21.0	1
22.0	1
24.5	1
28.0	1
27.0	0
20.1	1
25.6	0
27.2	1
29.0	1
28.2	1

데이터 셋 설명

- temp : 온도

- dog.panting (0: 헐떡임, 1: 헐떡이지 않음)

In [9]:
```
plot(dog.panting ~ temp)
```



로지스틱 회귀 모델 만들기

In [11]:
```
model <- glm(dog.panting~temp, data=data,
                  family = 'binomial')
dim(data)
summary(model)
```

20 2

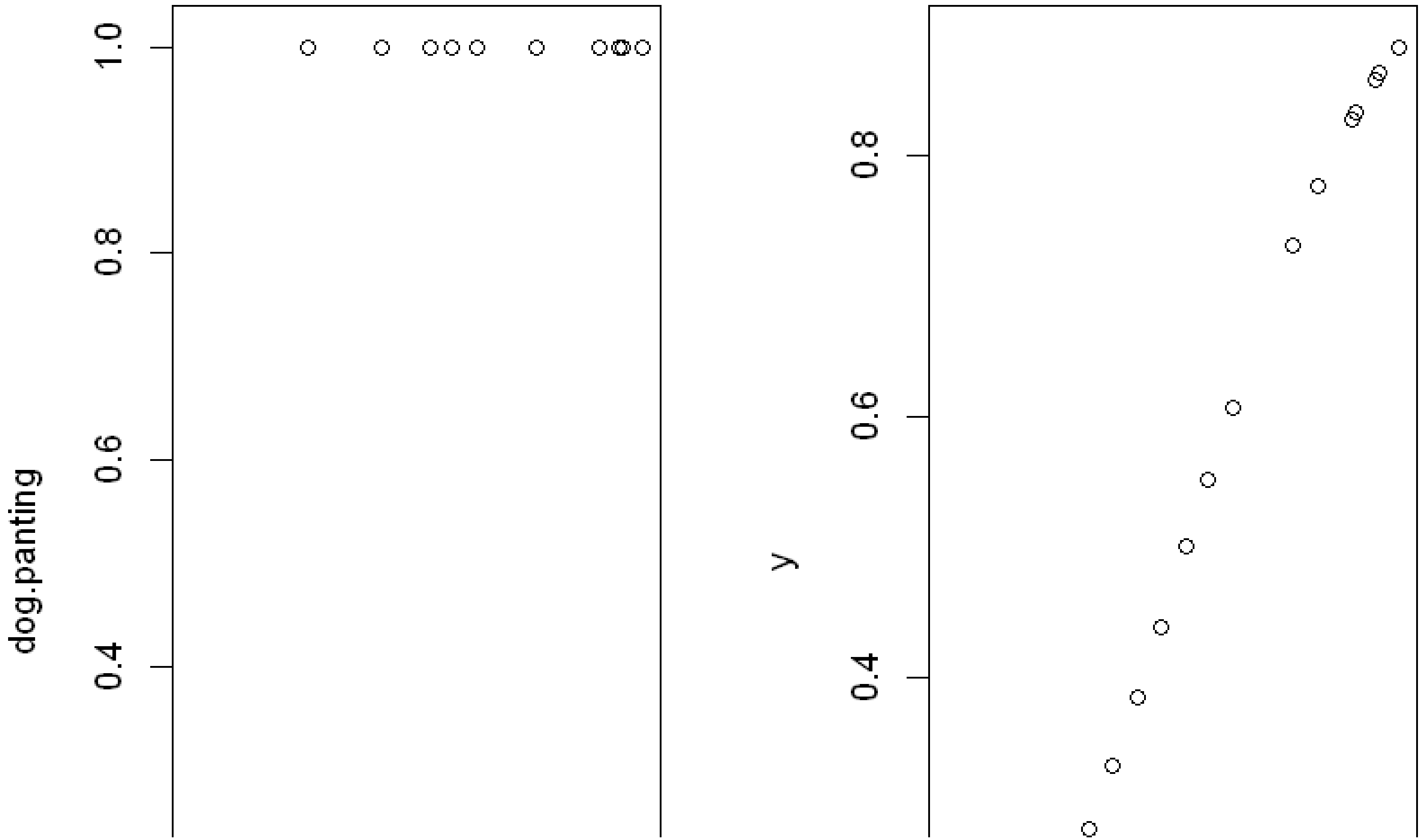
```
Call:
glm(formula = dog.panting ~ temp, family = "binomial", data = data)

Deviance Residuals: 
     Min        1Q    Median        3Q       Max  
-1.87261  -0.76187   0.02904   0.84385   1.69679  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)  
(Intercept)  -4.5334     2.1439  -2.115   0.0345 *
temp          0.2258     0.1042   2.167   0.0302 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 27.726  on 19  degrees of freedom
Residual deviance: 21.339  on 18  degrees of freedom
AIC: 25.339

Number of Fisher Scoring iterations: 3
```

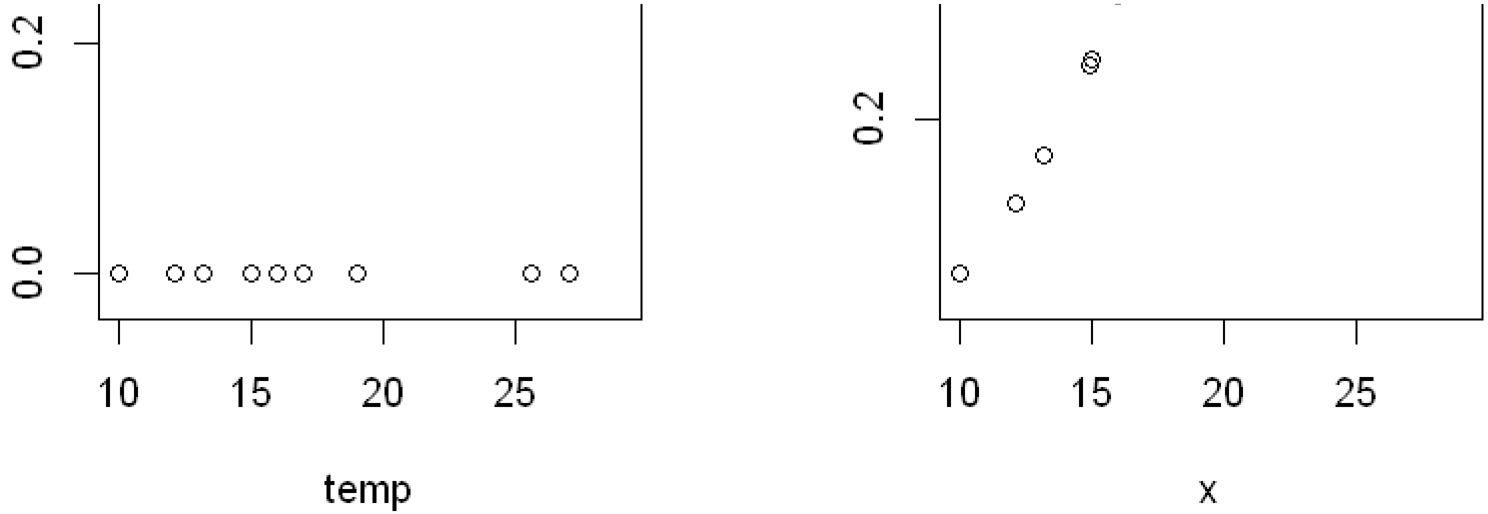
예측값과 잔차들 그래프

In [12]:
```
model$coefficients
```

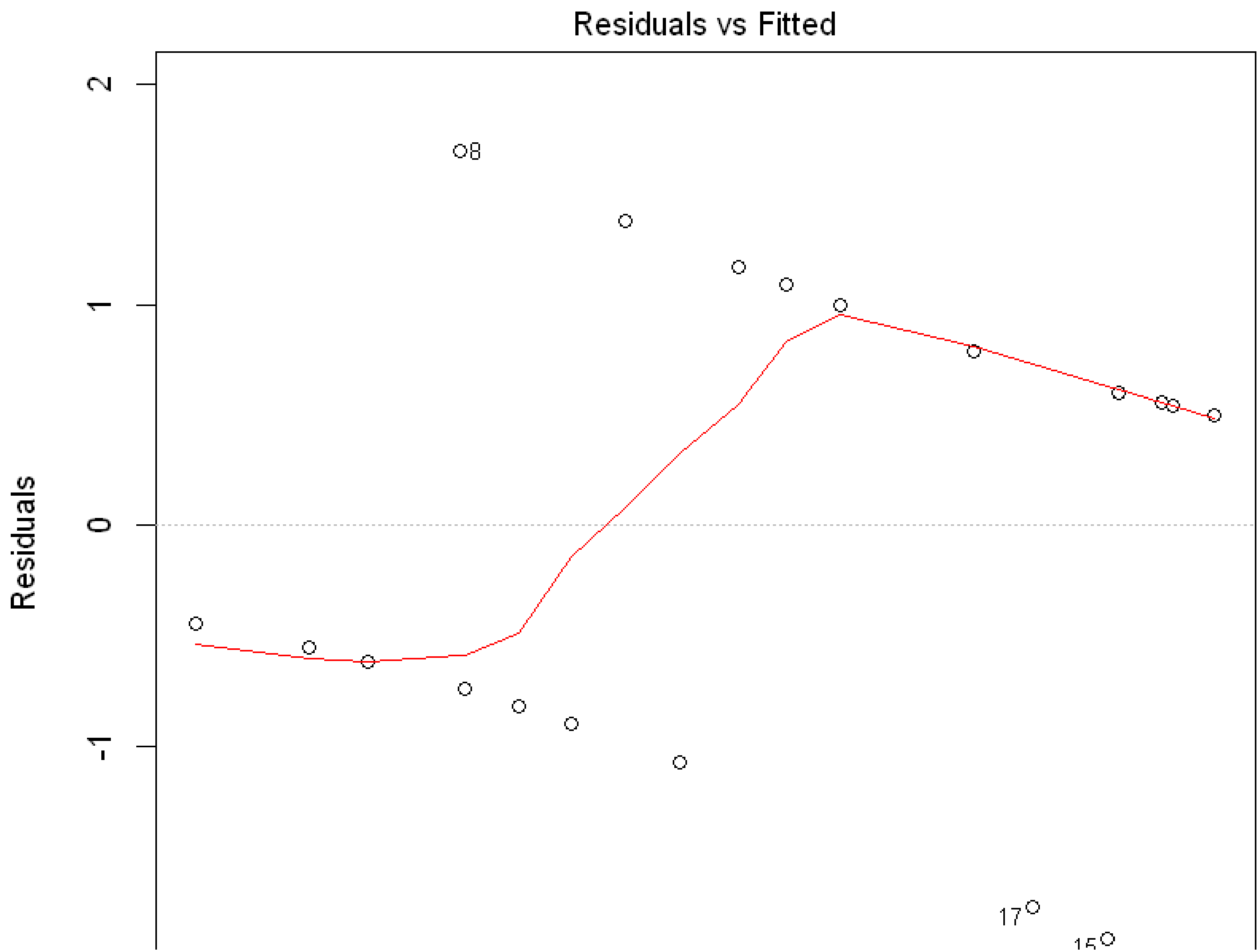
(Intercept)	-4.53338990823194
temp	0.225797708736263

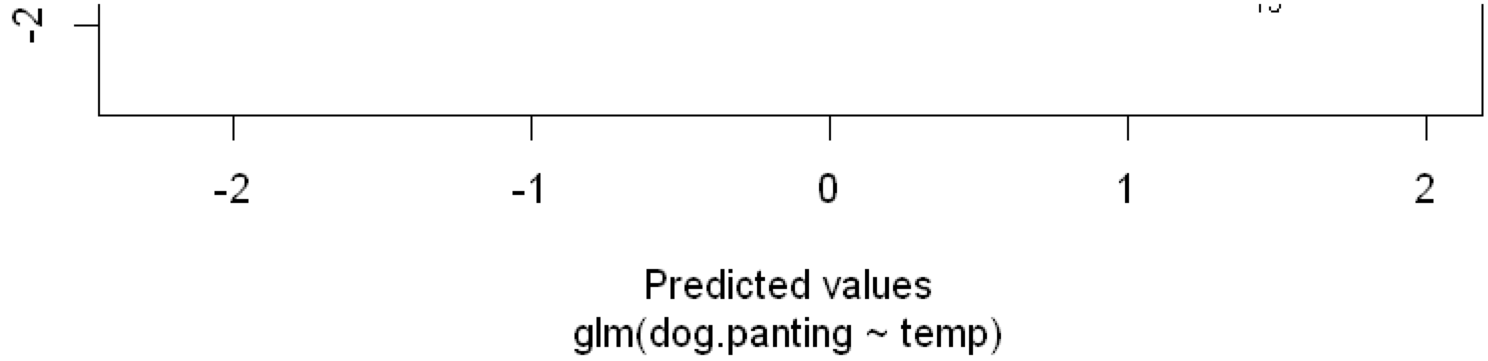
In [15]:
```
x <- temp
b <- model$coefficients[1]  # intercept
m <- model$coefficients[2]  # slope
y = exp((b + m*x)) / (1 + exp((b+ m*x)))
par(mfrow=c(1,2))
plot(temp, dog.panting)
plot(x,y)
```

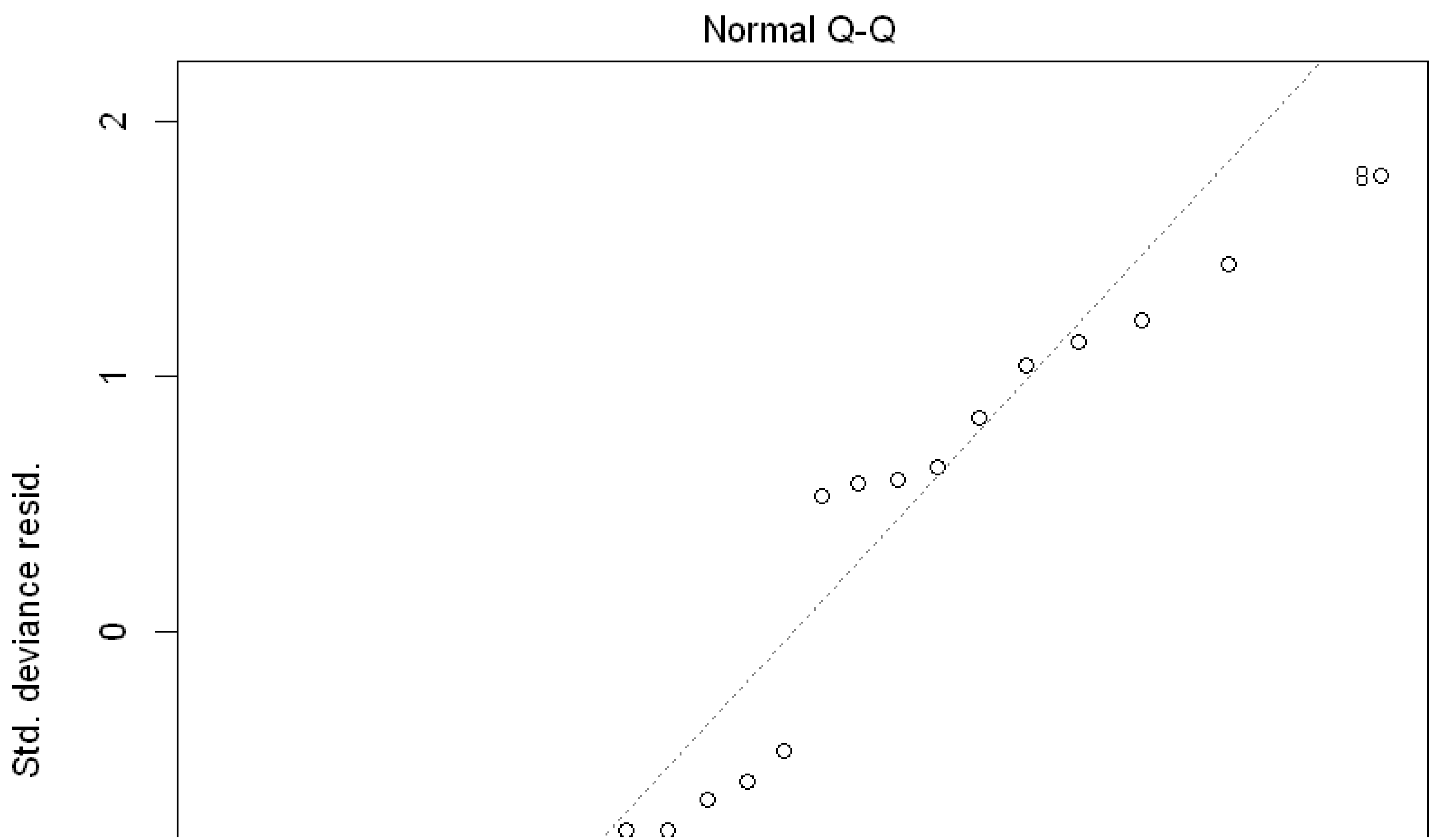


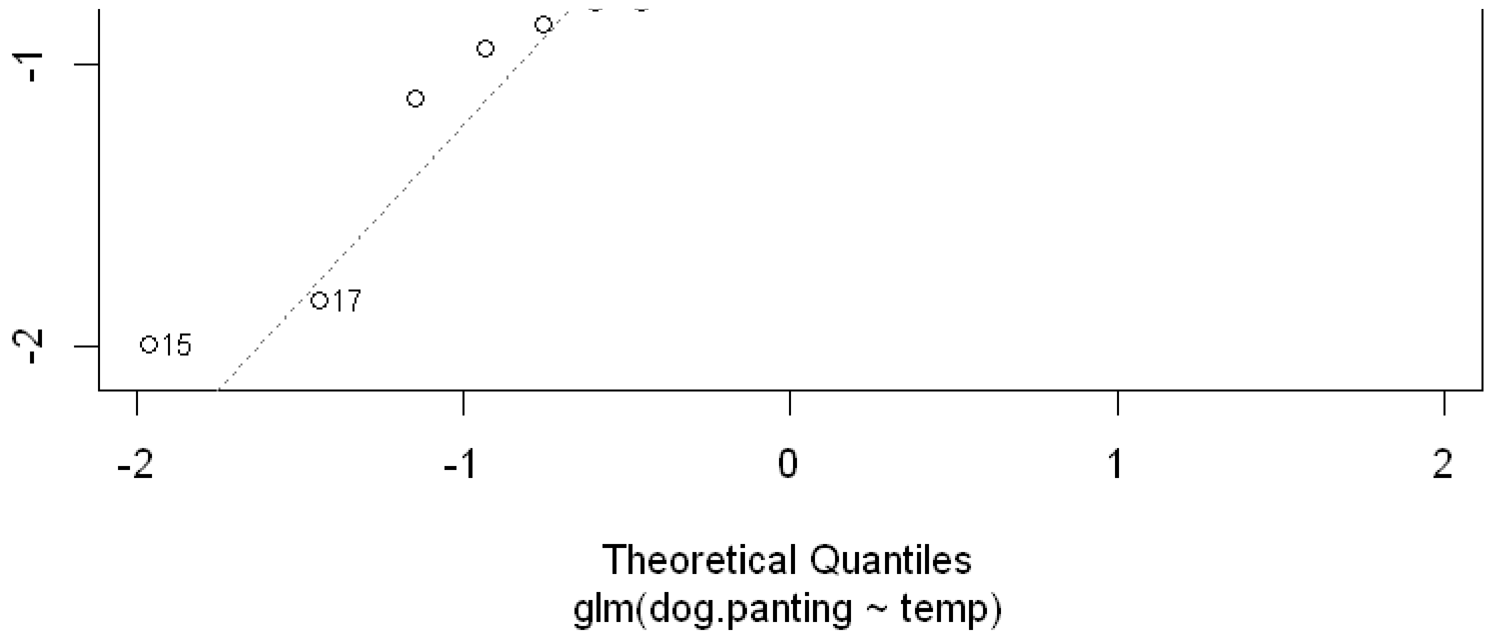

0.2
0.0
10
15
20
25
temp
0.2
10
15
20
25
x

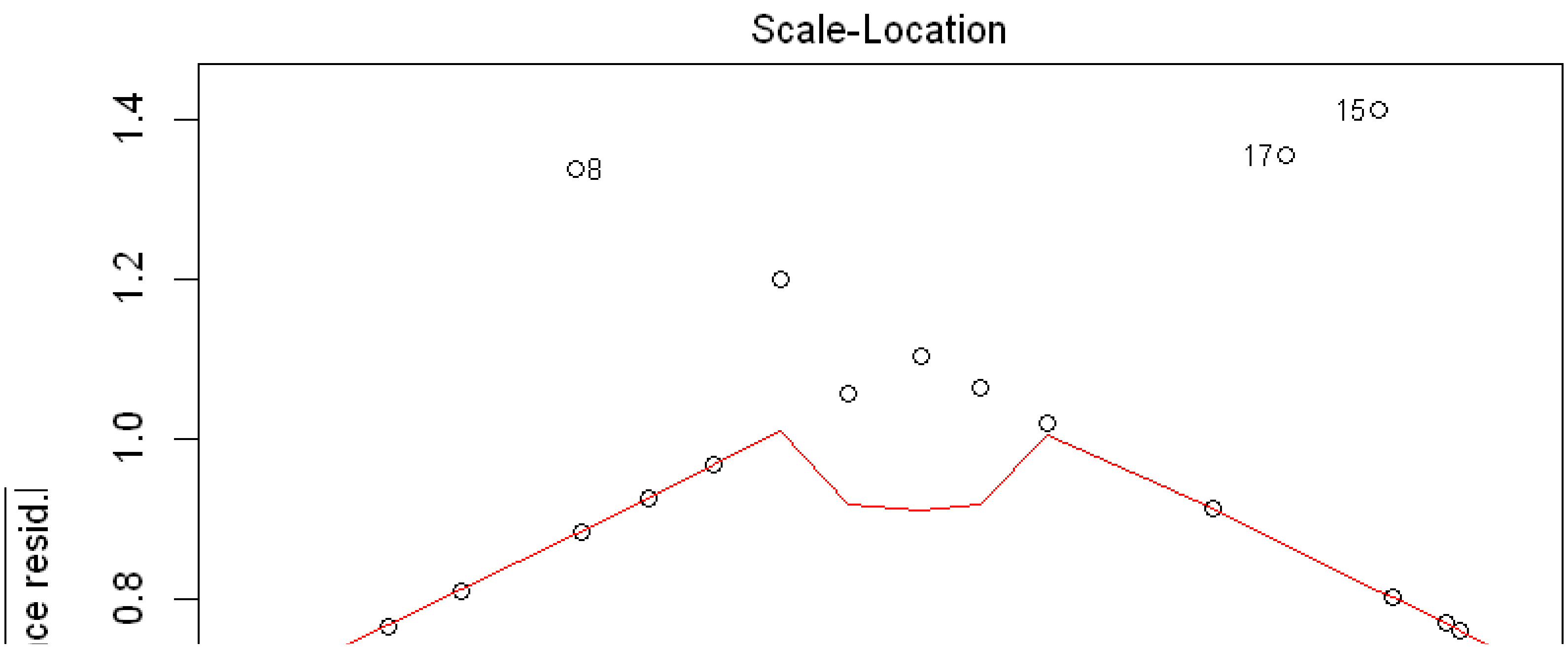
In [14]:
```
plot(model)
```

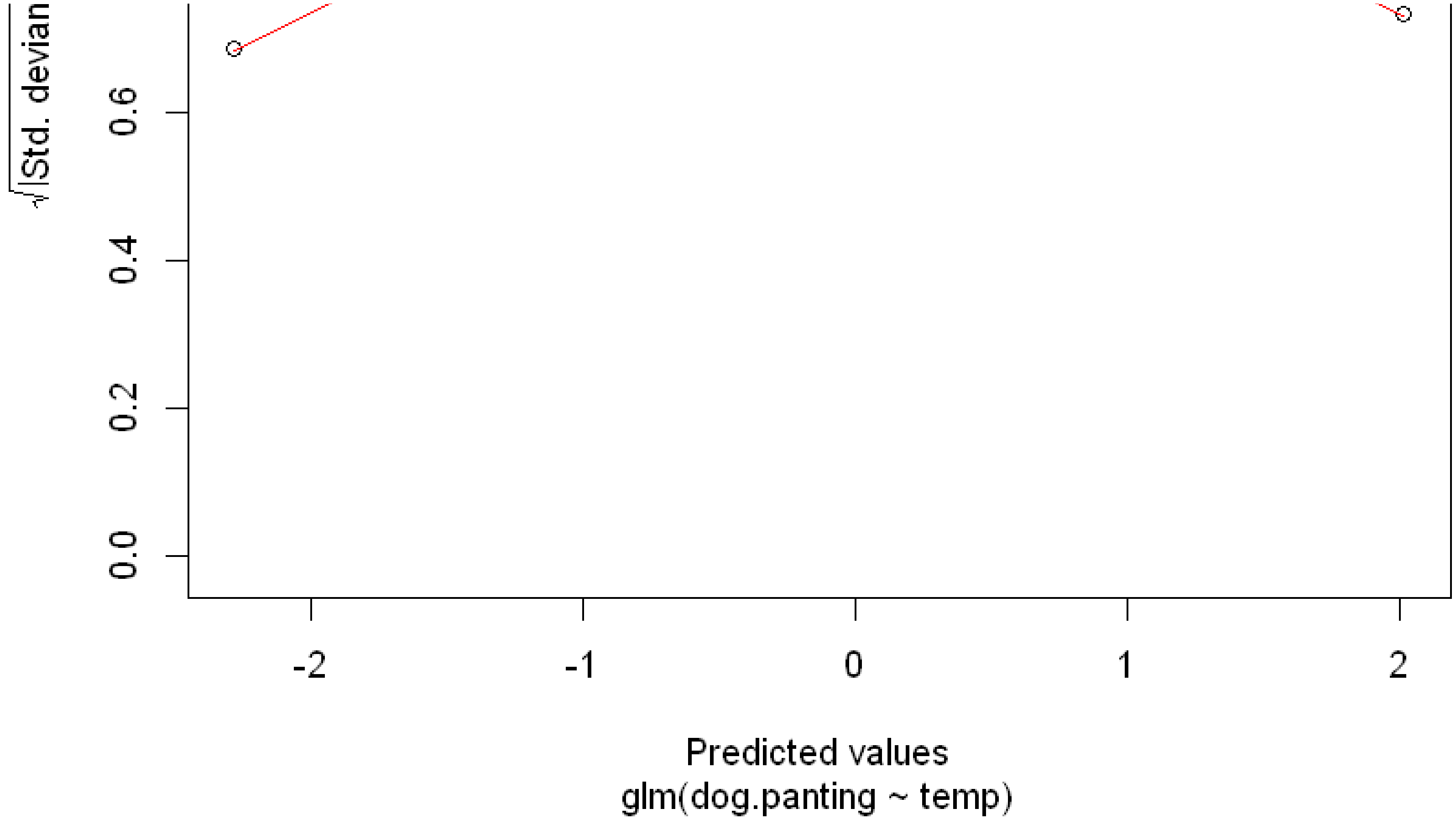


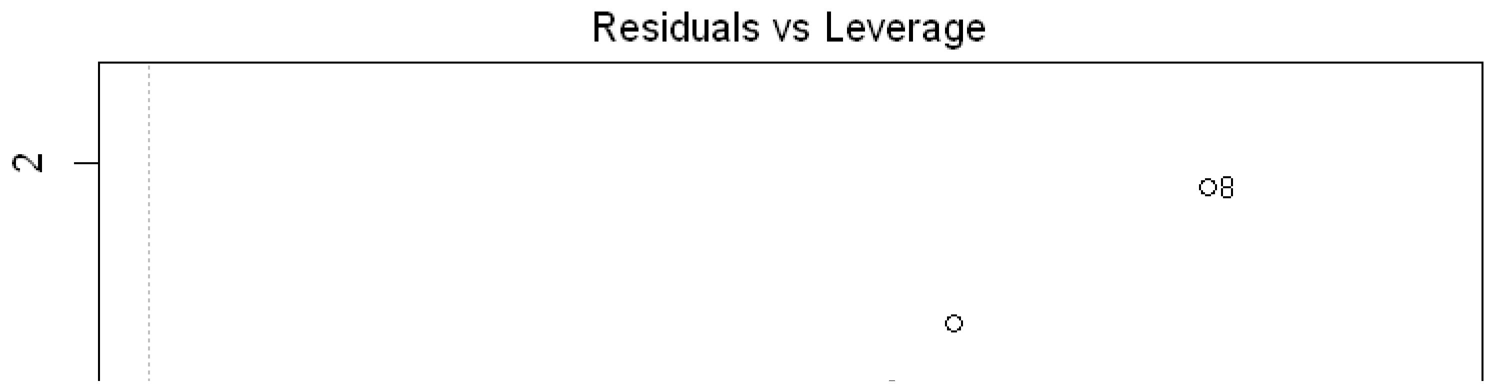

-2
-2
-1
0
1
2
Predicted values
glm(dog.panting ~ temp)

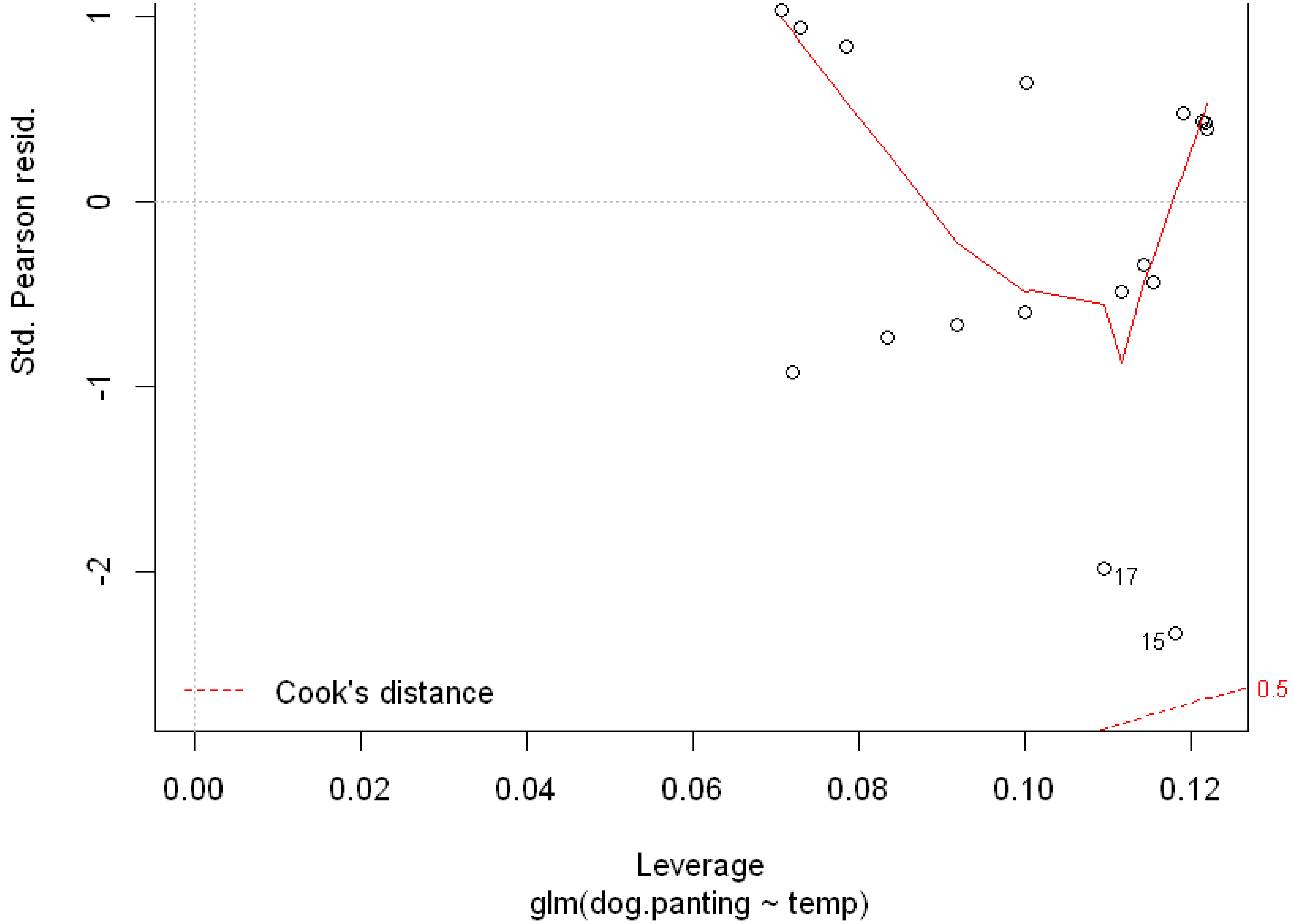

Normal Q-Q
Std. deviance resid.
2
1
0
8


-1
-2
15
17
-2
-1
0
1
2
Theoretical Quantiles
glm(dog.panting ~ temp)


Scale-Location
1.4
1.2
1.0
0.8
ce resid.|
8
17
15


√|Std. devian
0.0
0.2
0.4
0.6
-2
-1
0
1
2
Predicted values
glm(dog.panting ~ temp)


Residuals vs Leverage
2
8



로지스틱 회귀 분석 예

- mtcars 데이터 셋은 1974년 Motor Trend US magazine의 데이터 셋이다.
- 32종류의 자동차를 10가지 항목을 조사한 데이터 셋이다.
- 총 10가지 항목은 다음과 같다.

```
mpg : 연비(Miles/ (US) gallon
cyl : 실린더 개수(Number of cylinders)
disp : 배기량 (Displacement (cu.in))
hp   : 마력(Gross horespower)
drat : 후방차축 비율(Rear axle ratio)
wt   : 무게(Weight (1,000 lbs))
qsec : 1/4 마일에 도달하는데 소요되는 시간(1/4mile time)
vs   : 엔진(0 = V engine, 1=S engine)
am   : 변속기(0=자동, 1=수동)
gear : 기어 개수(Number of forward gears)
carb : 기화기 개수(Number of carburetors)
```

실습 1

- (1) data(mtcars)를 불러온다.
- (2) subset 또는 select, 또는 기본 열 선택을 이용하여 mpg, am, vs 열을 선택한다.
- (3) glm 함수를 이용하여 로지스틱 회귀 분석을 수행해보자. (vs ~ mpg)
- (4) summary ( )함수를 이용하여 AIC 통계량을 확인해 보자.

In [17]:

	mpg	am	vs
Mazda RX4	21.0	1	0
Mazda RX4 Wag	21.0	1	0
Datsun 710	22.8	1	1
Hornet 4 Drive	21.4	0	1
Hornet Sportabout	18.7	0	0
Valiant	18.1	0	1

In [19]:

```

```

```
Call:
glm(formula = vs ~ mpg, family = "binomial", data = dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max  
-2.2127  -0.5121  -0.2276   0.6402   1.6980  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)   
(Intercept)  -8.8331     3.1623  -2.793  0.00522 **
mpg           0.4304     0.1584   2.717  0.00659 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.860  on 31  degrees of freedom
Residual deviance: 25.533  on 30  degrees of freedom
AIC: 29.533

Number of Fisher Scoring iterations: 6
```

In [ ]:

```

```