Model selection and logistic regression

Regression Analysis

STAT5002

The University of Sydney

May 2025

THE UNIVERSITY OF SYDNEY

Regression Analysis

Topic 13: Multiple linear regression

Topic 14: Model selection

Topic 15: Logistic regression

Outline

Today:

- The general F-test
- Model selection
- Logistic regression

The general F-test

Last week: air pollution example

The data frame `environmental` has four environmental variables taken in New York City from May to September of 1973:

- `ozone` concentration (part per billion), solar `radiation` (langley), maximum daily `temperature` (Fahrenheit) and `wind` speed (mile per hour)

```
data("environmental", package = "lattice")
dim(environmental)
```

```
[1] 111   4
```

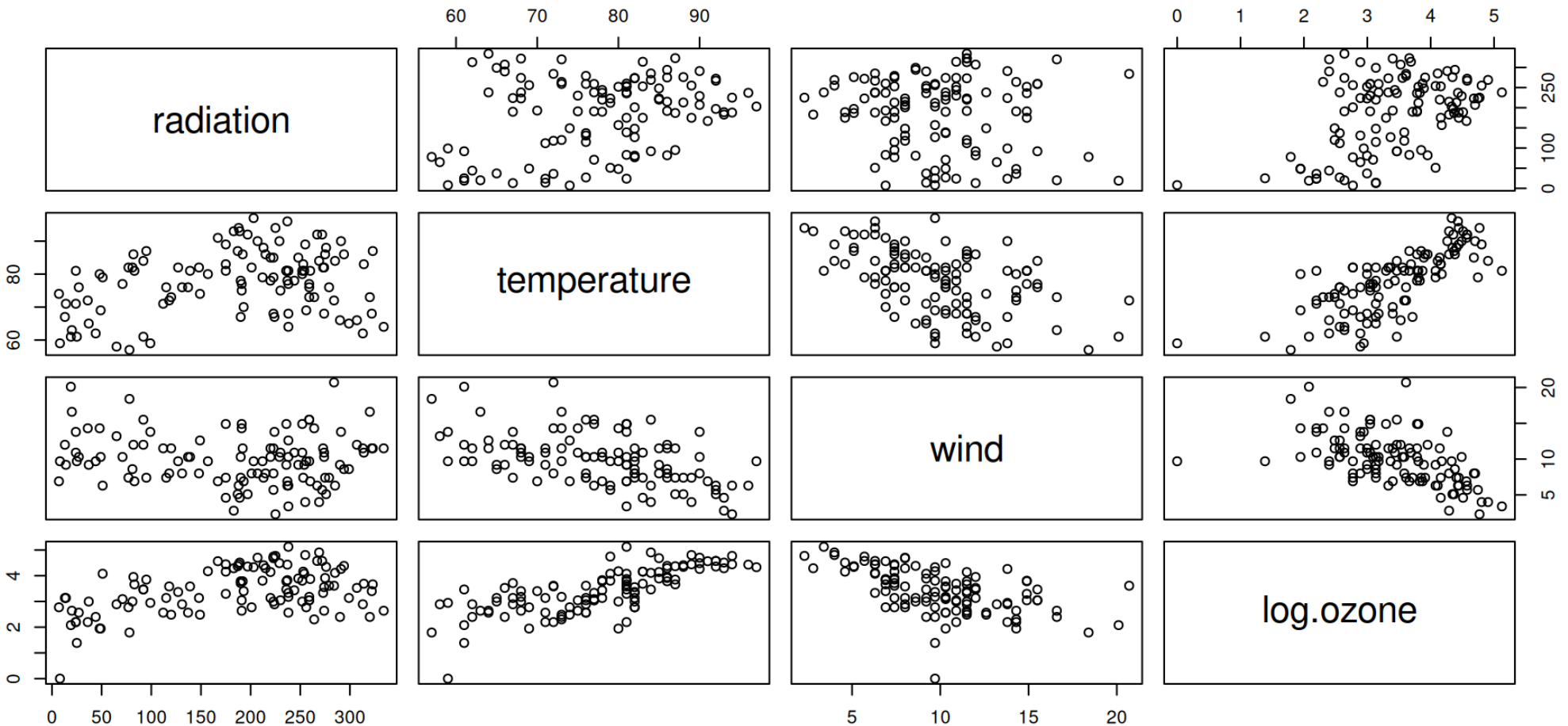
```
str(environmental)
```

```
'data.frame':   111 obs. of  4 variables:
 $ ozone      : num  41 36 12 18 23 19 8 16 11 14 ...
 $ radiation  : num  190 118 149 313 299 99 19 256 290 274 ...
 $ temperature: num  67 72 74 62 65 59 61 69 66 68 ...
 $ wind       : num  7.4 8 12.6 11.5 8.6 13.8 20.1 9.7 9.2 10.9 ...
```

- To remove nonlinearity, we considered the log of ozone concentration.

```
env.new = environmental  # create a new data frame
env.new[, "log.ozone"] = log(environmental$ozone)  # add a new variable log.ozone
env.new[, "ozone"] = NULL  # delete the old variable ozone
```

```r
pairs(env.new)
```



- The variable `log.ozone` appears to be positively associated with `temperature`, negatively associated with `wind`, and (moderately) positively associated with `radiation`.

Last week: multiple linear model

We have seen the multiple linear model

$$Y_i = b_0 + b_1 \cdot x_{1,i} + b_2 \cdot x_{2,i} + \ldots + b_p \cdot x_{p,i} + \varepsilon_i, \text{ where } \varepsilon_i \sim (\text{iid}) \ N(0, \sigma^2),$$

that incorporates multiple independent (explanatory) variables to predict the dependent variable Y.

```
lm3 = lm(log.ozone ~ radiation + temperature + wind, env.new)
round(summary(lm3)$coefficients, 3)
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.261      0.553  -0.472    0.638
radiation      0.003      0.001   4.518    0.000
temperature    0.049      0.006   8.078    0.000
wind          -0.062      0.016  -3.922    0.000
```

Fitted model for predicting log ozone:

$$\widehat{\log(\text{ozone})} = -0.261 + 0.003 \cdot \text{radiation} + 0.049 \cdot \text{temperature} - 0.062 \cdot \text{wind}$$

On average, a one degree increase in temperature results in an approximate 4.9% increase in ozone (increase of log ozone by 0.049 units), holding radiation and wind speed constant.

Last week: T-test in regression

Each estimated regression coefficient `Estimate` has an estimated standard error (`Std. Error`), and an oberserved T-statistic (`t value`).

- The T-test here aims to test whether independent variable x_j has a significant linear relationship with the dependent variable Y, after adjusting for all other independent variables in the model. For example,
  - $H_0 : b_2 = 0$ – after adjusting for all other independent variables, there is no linear relationship between `temperature` and `log.ozone`
  - $H_0 : b_2 \neq 0$ – after adjusting for all other independent variables, there is a linear relationship between `temperature` and `log.ozone`
  - In this case, we reject H_0 at the default 5% level of significance, as the P-value is much smaller than 5%.

We sometimes want to test whether the entire model (or a group of independent variables) explains a significant amount of variability in the dependent variable Y.

- T-test may not be suitable, because it applies to one independent variable at a time.
- We can use the **F-test** for this.

F-test

The F-test is used to assess two nested models, where

- the **null model** is a special case of a more complicated **alternative model** containing additional independent variables.
  - ➠ That is, the (reduced) null model contains a subset of independent variables of the (larger) alternative model.
  - ➠ From example, some of the possible models for the air pollution data are

$$\begin{aligned}
\textbf{Model 4:} \quad & \log(\text{ozone}_i) = b_0 + b_1 \cdot \text{radiation}_i + b_2 \cdot \text{temperature}_i + b_3 \cdot \text{wind}_i + \varepsilon_i \\
\textbf{Model 3:} \quad & \log(\text{ozone}_i) = b_0 + b_1 \cdot \text{radiation}_i + b_2 \cdot \text{temperature}_i + \varepsilon_i \\
\textbf{Model 2:} \quad & \log(\text{ozone}_i) = b_0 + b_2 \cdot \text{temperature}_i + \varepsilon_i \\
\textbf{Model 1:} \quad & \log(\text{ozone}_i) = b_0 + \varepsilon_i
\end{aligned}$$

    - ➠ When Model 1 is the null model, Model 2, 3, or 4 can be a valid alternative model
    - ➠ When Model 2 is the null model, Model 3 or 4 can be a valid alternative model
- We may want to test whether the additional independent variables in the alternative model significantly improve the fit of the null model.
- A The F-test relies on the same set of assumptions as multiple linear regression.

H the overall test

- $H_0 : b_1 = b_2 = b_3 = 0$
  - All regression coefficients (except the intercept) are zero.
  - None of the independent variables help explain the dependent variable.
  - This corresponds to the null model

$$Y_i = b_0 + \varepsilon_i.$$

- H_1 **:** at least one of the regression coefficients ($b_1, \ldots, b_p$) is not zero.
  - At least one of the independent variable has an effect in explaining the dependent variable.
  - This corresponds to the alternative model

$$Y_i = b_0 + b_1 \cdot x_{1,i} + b_2 \cdot x_{2,i} + b_3 \cdot x_{3,i} + \varepsilon_i.$$

H the partial test

- $H_0 : b_1 = b_3 = 0$: The additional independent variables x_1 (`radiation`) and x_3 (`wind`) have no effect in explaining Y (`log.ozone`).
  - ➠ This corresponds to the null model

$$Y_i = b_0 + b_2 \cdot x_{2,i} + \varepsilon_i.$$

- H_1 **:** at least one of the additional independent variables, x_1 or x_3, has an effect in explaining Y.
  - ➠ This corresponds to the alternative model

$$Y_i = b_0 + b_1 \cdot x_{1,i} + b_2 \cdot x_{2,i} + b_3 \cdot x_{3,i} + \varepsilon_i.$$

- This can be applied to other nested models.

Consider a null model with q independent variables and an alternative model with p independent variables. The alternative model is always larger, so $p > q$.

- Under H_0:
  - Fit the null model and calculate $\widehat{SSE}_{H_0}$ measuring unexplained variation of the null model.
  - The degrees of freedom is $n - (q + 1)$.
- Under H_1:
  - Fit the alternative model and calculate $\widehat{SSE}_{H_1}$ measuring its unexplained variation.
  - The degrees of freedom is $n - (p + 1)$.
- The F-statistic:

$$F = \frac{(\widehat{SSE}_{H_0} - \widehat{SSE}_{H_1})/(p - q)}{\widehat{SSE}_{H_1}/\left(n - (p + 1)\right)} \sim F_{p-q,n-(p+1)}.$$

Remarks on the F-statistic

The F-statistic:

$$F = \frac{(\widehat{SSE}_{H_0} - \widehat{SSE}_{H_1})/(p-q)}{\widehat{SSE}_{H_1}/\left(n-(p+1)\right)} \sim F_{p-q,n-(p+1)}.$$

- $F_{p-q,n-(p+1)}$: F-distribution (we skip its detail here).
- Numerator: **explained variation per additional independent variable**.
  - $F > 0$, as the SSE of the alternative model is always less than that of the null model (with additional independent variables), $\widehat{SSE}_{H_1} \leq \widehat{SSE}_{H_0}$
  - $\widehat{SSE}_{H_0} - \widehat{SSE}_{H_1}$ gives the variations of the null model that can be explained by the alternative
  - $(p-q)$ is the number of additional independent variables of the alternative model
- Denominator: **unexplained variation in the alternative model per degree of freedom**
  - The denominator is also the estimated variance of the error box of the alternative model.
- One-sided test, only large values of F argue against H_0.
- The F-statistic provided in `summary(lm( ... ))` is the statistic for the overall test.

Example

Consider the overall test for air pollution data.

H

- $H_0 : b_1 = b_2 = b_3 = 0$
  - ➟ None of the independent variables help explain the dependent variable.
- H_1 : at least one of the regression coefficients is not zero.
  - ➟ At least one of the independent variable has an effect in explaining the dependent variable.

T Let's verify the F-statistic "by hand"

```
sse.alternative = sum(lm3$residuals^2)
sse.null = sum((env.new$log.ozone - mean(env.new$log.ozone))^2)  # the null model is the sample mean of y
p = 3
q = 0
n = length(env.new$log.ozone)
numer = (sse.null - sse.alternative)/(p - q)
est.var = sse.alternative/(n - (p + 1))
f.stat = numer/est.var
round(f.stat, 2)
```

```
[1] 70.65
```

P Since it's a one-sided test, P-value $= P(F > f)$

```
1 pf(f.stat, p - q, n - (p + 1), lower.tail = F)
```

```
[1] 2.860349e-25
```

- Very small P-value.

```
1 summary(lm3)
```

```

Call:
lm(formula = log.ozone ~ radiation + temperature + wind, data = env.new)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.06212 -0.29968 -0.00223  0.30767  1.23572 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.2611739  0.5534102  -0.472 0.637934    
radiation    0.0025147  0.0005567   4.518 1.62e-05 ***
temperature  0.0491630  0.0060863   8.078 1.07e-12 ***
wind        -0.0615925  0.0157037  -3.922 0.000155 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5085 on 107 degrees of freedom
Multiple R-squared:  0.6645,    Adjusted R-squared:  0.6551 
F-statistic: 70.65 on 3 and 107 DF,  p-value: < 2.2e-16
```

Adjusted R-squared versus R-squared (coefficient of determination)

- Recall the coefficient of determination (`Multiple R-squared` in the output of `summary(lm( ... ))`)

$$\text{CoD} = 1 - \frac{\text{sum of squared residuals of the fitted model}}{\text{sum of squared deviations of the dependent variable}} = 1 - \frac{\widehat{SSE}}{\widehat{SST}}$$

- Adding variables tends to decrease $\widehat{SSE}$, and hence tends to increase CoD due to added complexity.

$$\text{Adjusted R-squared} = 1 - \frac{\text{Estimated SD of the residual error}}{\text{Sample SD of the dependent variable}} = 1 - \frac{\hat{\sigma}}{\hat{s}_Y}$$

- Adjusted R-squared penalizes the inclusion of unhelpful independent variables:

$$\text{Adjusted R-squared} = 1 - \frac{\widehat{SSE}/(n-(p+1))}{\widehat{SST}/(n-1)} = 1 - (1-\text{CoD})\frac{n-1}{n-(p+1)} \leq \text{CoD}$$

```
1 1 - (1 - summary(lm3)$r.squared) * (n - 1)/(n - (p + 1))  # sanity check
```

```
[1] 0.6551089
```

Expected learning outcomes

- Know how to apply F-test to compare two models
  - ➟ We don't expect you remember the test statistic of F-test
  - ➟ But you need to know what are the null and alternative hypotheses of F-test;
  - ➟ and how to interpret the given P-value to compare models.
- Know the difference between the coefficient of determination (R-squared) and the adjusted R-squared
  - ➟ We don't expect you remember the formula of the adjusted R-squared

Model selection

General thoughts on choosing between models

In choosing between models, statisticians have two aims:

- To choose a simple (i.e. not too complex) model
  - ➟ A possibility to measure the complexity of a linear regression model is by the number of independent variables, p. The greater this value, the more complex the model.
- To choose a model that fits the data well.
  - ➟ Possibilities to measure the closeness of fit of the model to data are R-squared, adjusted R-squared, etc.

Purpose of model selection is to choose one or more models that can balance the complexity with the goodness of fit.

- Think of model selection like shopping – is it worth spending more (independent variables) in order to get a better (fitting) model?
- We will discuss two iterative processes for model selection.

Backward variable selection

We start with a full model containing all possible independent variables. In each iteration of the backward variable selection:

1. Start with the current model, for each independent variable in turn, investigate the **effect** of removing a variable from the current model.
2. Remove the least significant variable, unless this independent variable is supplying **significant information** about the dependent variable Y.
3. Go to step 1. Stop only if all variables in the current model are important.

There are many ways to measure the "effect" and "significance". Possibilities include

- P-values of hypothesis tests. For example, F- or T- tests.
- R-squared and adjusted R-squared.
- Other information criteria (we skip those here).

We will use F-test here.

Example: cheese tasting

Data were collected from the production of cheddar cheese in the LaTrobe Valley of Victoria.

- $n = 30$ samples of cheese were tasted by experts;
- taste of the final product is related to the concentration of several chemicals in the cheese; and
- the following four variables recorded:
  - `taste`: Tasters' ratings
  - `Acetic`: Acetic acid in the cheese
  - `H2S`: Hydrogen sulphide in the cheese
  - `Lactic`: Lactic acid in the cheese.

We construct a multiple linear regression model (using backward selection) to investigate how chemical variables relate to the cheese's taste.

The least significant variable will be removed based on an F-test conducted at the 5% significance level.

- Load the cheese data

```r
dat = read.table(file = "./data/cheese.txt", header = TRUE, row.names = NULL)
str(dat)
```

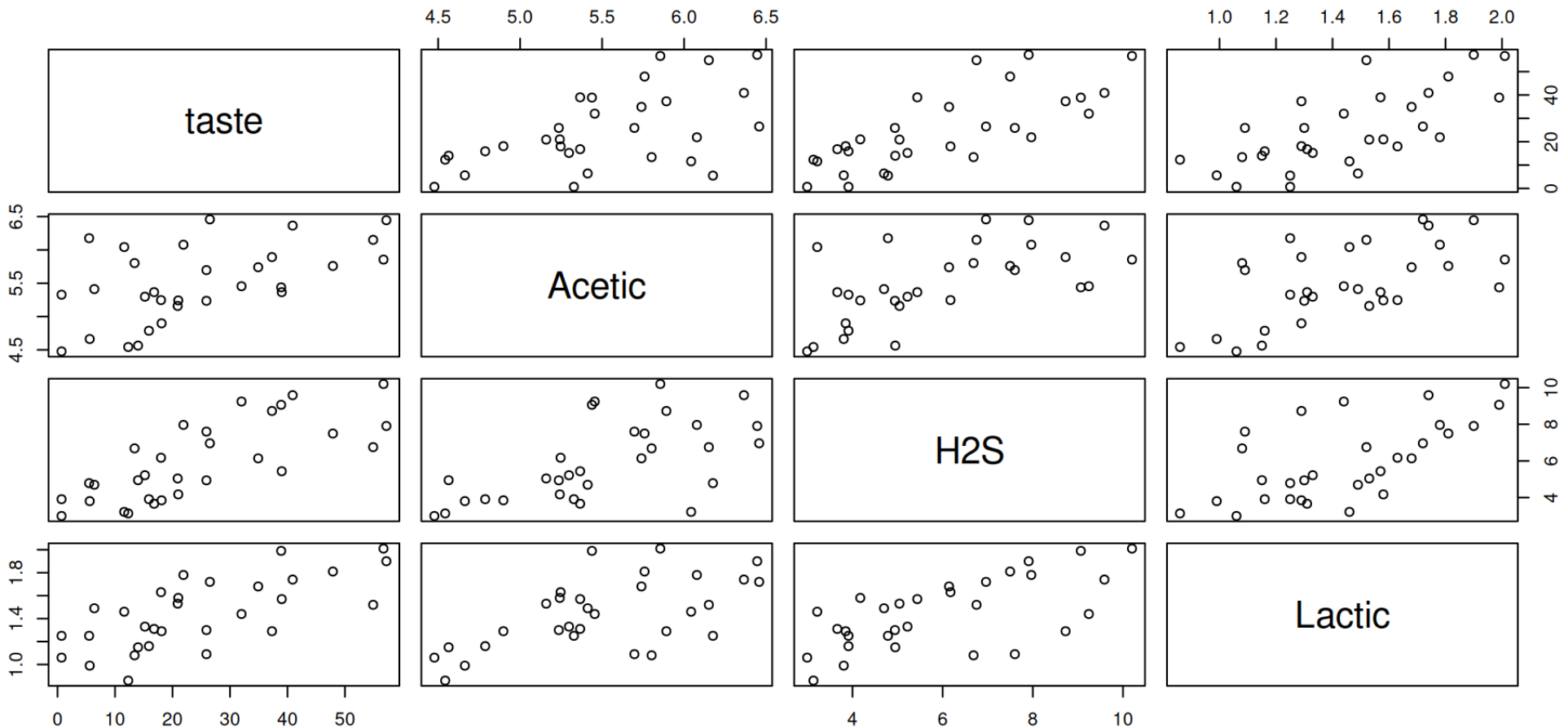
```
'data.frame':   30 obs. of  4 variables:
 $ taste : num  12.3 20.9 39 47.9 5.6 25.9 37.3 21.9 18.1 21 ...
 $ Acetic: num  4.54 5.16 5.37 5.76 4.66 ...
 $ H2S   : num  3.13 5.04 5.44 7.5 3.81 ...
 $ Lactic: num  0.86 1.53 1.57 1.81 0.99 1.09 1.29 1.78 1.29 1.58 ...
```

- Pairwise correlation coefficients

```r
round(cor(dat), 2)
```

```
       taste Acetic  H2S Lactic
taste   1.00   0.55 0.76   0.70
Acetic  0.55   1.00 0.62   0.60
H2S     0.76   0.62 1.00   0.64
Lactic  0.70   0.60 0.64   1.00
```

- Pairwise scatter plots



- All independent variables show positive linear associations with the dependent variable.
- The independent variables are also positively associated with each other, which may suggest multicollinearity.

Start with the full model

- `lm(taste ~ ., data = dat)`: `taste ~ .` includes all variables other than `taste` as the independent variable

```
M1 = lm(taste ~ ., data = dat)
summary(M1)
```

```
Call:
lm(formula = taste ~ ., data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-17.390  -6.612  -1.009   4.908  25.449

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -28.8768    19.7354  -1.463  0.15540
Acetic        0.3277     4.4598   0.073  0.94198
H2S           3.9118     1.2484   3.133  0.00425 **
Lactic       19.6705     8.6291   2.280  0.03108 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.13 on 26 degrees of freedom
Multiple R-squared:  0.6518,    Adjusted R-squared:  0.6116
F-statistic: 16.22 on 3 and 26 DF,  p-value: 3.81e-06
```

1st iteration: determine which variable to remove

- the function `drop1(M1, test = "F")` performs single-term deletions from the model `M1` and shows how the model fit changes after removing the independent variable.

```r
drop1(M1, test = "F")
```

```
Single term deletions

Model:
taste ~ Acetic + H2S + Lactic
       Df Sum of Sq    RSS    AIC F value   Pr(>F)
<none>              2668.4 142.64
Acetic  1      0.55 2669.0 140.65  0.0054 0.941980
H2S     1   1007.66 3676.1 150.25  9.8182 0.004247 **
Lactic  1    533.32 3201.7 146.11  5.1964 0.031079 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- `Df`: degrees of freedom dropped
- `Sum of Sq`: the increase in SSE of the reduced model, compared to the full model `M1`
- `RSS`: sum of squared errors (SSE) of the reduced model
- `F value` and `Pr(>F)`: for assessing the impact of removing the independent variable

Sanity checks

- See the following for the sanity check. We first build three reduced models

```
rm1 = lm(taste ~ . - Acetic, data = dat)
rm1$coefficients
```

```
(Intercept)         H2S      Lactic
 -27.591815    3.946267   19.887204
```

```
rm2 = lm(taste ~ . - H2S, data = dat)
rm2$coefficients
```

```
(Intercept)      Acetic      Lactic
  -51.36603     5.57139    31.39229
```

```
rm3 = lm(taste ~ . - Lactic, data = dat)
rm3$coefficients
```

```
(Intercept)      Acetic         H2S
 -26.939727    3.801199    5.145598
```

- The syntax `. - Acetic` stands for removing `Acetic` from the existing included variables.

- Calculate the SSEs of reduced models, compare them with that of the full model `M1`

```r
sse.rm = c(sum(rm1$residuals^2), sum(rm2$residuals^2), sum(rm3$residuals^2))  # SSE of reduced models
sse.full = sum(M1$residuals^2)  # sse of M1
sse.increase = sse.rm - sse.full
round(rbind(sse.rm, sse.increase), 2)
```

```
               [,1]    [,2]    [,3]
sse.rm       2668.97 3676.07 3201.73
sse.increase    0.55 1007.66  533.32
```

- The F-test takes each of the reduced models as the null model and use the full model `M1` as the alternative. See the following for the sanity check

```r
n = 30
p = 3
q = 2
numer = (sse.rm - sse.full)/(p - q)
est.var = sse.full/(n - (p + 1))
f.stat = numer/est.var
round(f.stat, 4)
```

```
[1] 0.0054 9.8182 5.1964
```

1st iteration: model update

- Calculate the P-value (also given in the output of `drop1`)

```
pf(f.stat, p - q, n - (p + 1), lower.tail = F)
```

```
[1] 0.941979774 0.004247081 0.031079481
```

- If the P-value is small (e.g., < 0.05):
  - ➟ The increase in SSE is large.
  - ➟ The independent variable contributes significantly to the model, so we reject H_0 that this variable has no effect.
- Otherwise (e.g., P-value ≥ 0.05):
  - ➟ The increase in SSE is small.
  - ➟ We fail to reject H_0. The independent variable may not be necessary in the model.
  - ➟ So we can remove it (the one with largest P-value) from the model.
  - ➟ Remove `Acetic` in this case.

```
M2 = update(M1, . ~ . - Acetic, data = dat)
```

- The function `M2 = update(M1, . ~ . - Acetic, data = dat)` updates the model `M1` by deleting the variable `Acetic`
  - ➟ The syntax `. ~ .` stands for "whatever was in the corresponding position in the old formula"

2nd iteration

```r
drop1(M2, test = "F")
```

```
Single term deletions

Model:
taste ~ H2S + Lactic
       Df Sum of Sq    RSS    AIC F value   Pr(>F)
<none>              2669.0 140.65
H2S     1   1193.52 3862.5 149.74 12.0740 0.001743 **
Lactic  1    617.18 3286.1 144.89  6.2435 0.018850 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Since all the P-values are < 0.05, indicating the remaining independent variables contribute significantly to the model, so we stop.

```r
round(M2$coefficients, 2)
```

```
(Intercept)         H2S      Lactic
     -27.59        3.95       19.89
```

- Hence, the "best" model for the data (according to backward selection with the significance level 5%) is

$$\widehat{\text{taste}} = -27.59 + 3.95 \cdot \text{H2S} + 19.89 \cdot \text{Lactic}$$

Forward variable selection

We start with the model containing no independent variables, i.e., the baseline model $\hat{y} = \bar{y}$. In each iteration of the forward variable selection:

1. For each variable in turn, investigate the effect of adding an independent variable to the current model.
2. Add the **most informative** variable, unless this variable is not supplying **significant information** about the dependent variable Y.
3. Go to step 1. Stop only if all of the non-included variables are not significant.

Similar to the backward selection, we will use F-test to measure the "effect" and "significance".

The baseline model:

```
1 M1 = lm(taste ~ 1, data = dat)
```

- `taste ~ 1`: the model only contains the intercept

1st iteration: determine which variable to add

- `add1(M1, scope = ~ Acetic + H2S + Lactic, test="F")` shows how the model fit changes after adding an independent variable not currently in the existing model `M1`.
    - ➠ `scope` gives the set of variables we want to consider adding.

```
1 add1(M1, scope = ~Acetic + H2S + Lactic, test = "F")
```

```
Single term additions

Model:
taste ~ 1
       Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>              7662.9 168.29
Acetic  1    2314.1 5348.7 159.50  12.114  0.001658 **
H2S     1    4376.7 3286.1 144.89  37.293 1.374e-06 ***
Lactic  1    3800.4 3862.5 149.74  27.550 1.405e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- `Df`: degrees of freedom added
- `Sum of Sq`: the **decrease** in SSE after adding the variable, compared to the current model `M1`
- `RSS`: sum of squared errors (SSE) of the larger model after adding the variable
- `F value` and `Pr(>F)`: for assessing the impact of adding the independent variable

Model update

- If the P-value is small (e.g., < 0.05):
  - ➠ The decrease in SSE is large.
  - ➠ The independent variable contributes significantly to the model, so we reject H_0 that this variable has no effect.
  - ➠ We add the independent variable with the smallest P-value
- Otherwise (e.g., P-value ≥ 0.05):
  - ➠ The decrease in SSE is small.
  - ➠ We fail to reject H_0. The independent variable may not be necessary in the model.
- Should add `H2S` to the current model

```
M2 = update(M1, . ~ . + H2S)
```

2nd iteration

```r
add1(M2, scope = ~Acetic + H2S + Lactic, test = "F")
```

```
Single term additions

Model:
taste ~ H2S
       Df Sum of Sq    RSS    AIC F value  Pr(>F)
<none>              3286.1 144.89
Acetic  1     84.41 3201.7 146.11  0.7118 0.40625
Lactic  1    617.18 2669.0 140.65  6.2435 0.01885 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- `scope = ~ Acetic + H2S + Lactic` and `scope = ~ Acetic + Lactic` give the same result, as `H2S` is already in `M2`
- Should add `Lactic` to the current model

```r
M3 = update(M2, . ~ . + Lactic)
```

3rd iteration

```
1 add1(M3, scope = ~Acetic + H2S + Lactic, test = "F")
```

```
Single term additions

Model:
taste ~ H2S + Lactic
       Df Sum of Sq    RSS    AIC F value Pr(>F)
<none>              2669.0 140.65
Acetic  1   0.55427 2668.4 142.64  0.0054  0.942
```

- Since all the P-values are ≥ 0.05, indicating the remaining independent variable may not be necessary in the model, so we stop.

```
1 round(M3$coefficients, 2)
```

```
(Intercept)         H2S      Lactic
     -27.59        3.95       19.89
```

- We end up with the same model as the backward selection.

Summary

In backward selection, we start with the full model

- We remove the statistically least significant variable (with the largest P-value, or equivalently, the smallest F-statistic) from the current model in each iteration.
- Until the P-value of all the remaining variables in the model, which are shown by `drop1()`, are < 0.05.

In forward selection, we start with the full model

- We add the statistically most significant variable (with the smallest P-value, or equivalently, the largest F-statistic) to the current model in each iteration.
- Until the P-values of all the remaining variables that are not in the model, which are shown by `add1()`, are ≥ 0.05.

Logistic regression

Model chance (probability) as a dependent variable

A dataset containing information from 25 student groups was collected. For each group, the dataset records:

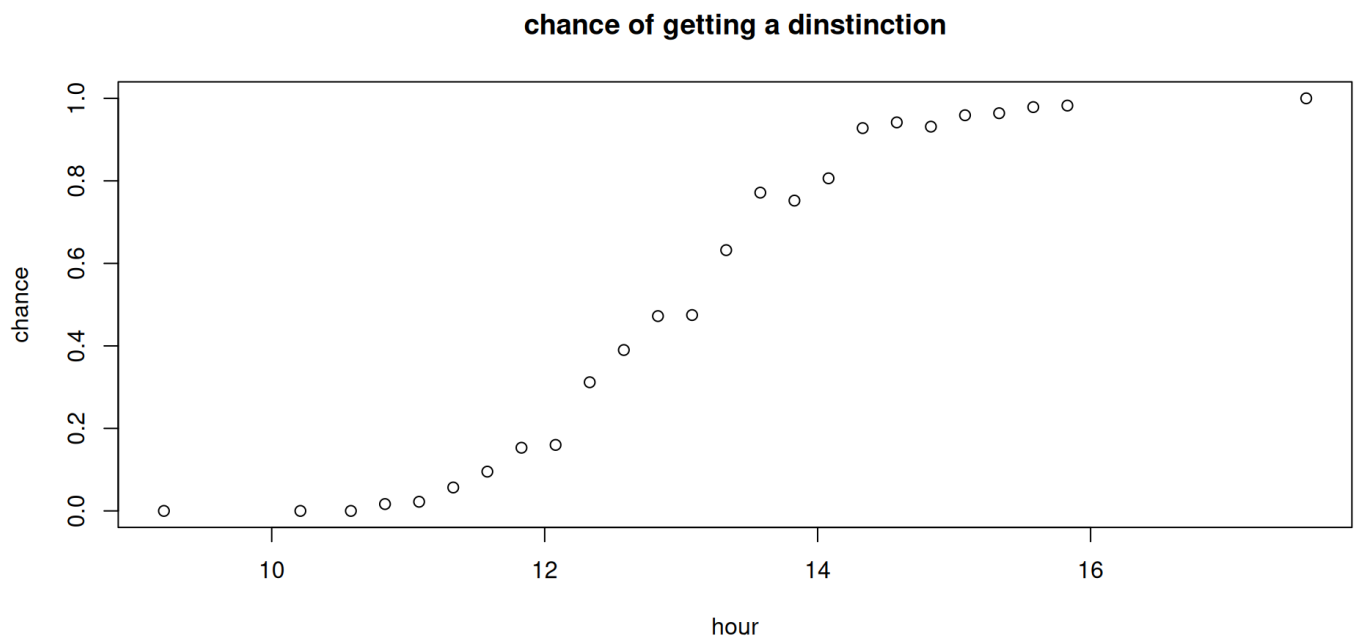
- `hour`: the average number of study hours per week of the group;
- `distinction`: the number of students who achieved the “Distinction” grade; and
- `total`: the total number of students in the group.

```r
dat = read.csv("data/units.csv", header = T)
str(dat)
```

```
'data.frame':   25 obs. of  3 variables:
 $ hour       : num  9.21 10.21 10.58 10.83 11.08 ...
 $ total      : int  376 200 93 120 90 88 105 111 100 93 ...
 $ distinction: int  0 0 0 2 2 5 10 17 16 29 ...
```

- `distinction/total` gives the proportion of students who have achieved the "Distinction" grade in each group
    - ➠ it can be interpreted as a chance

```r
distinction = dat$distinction
total = dat$total
chance = distinction/total
hour = dat$hour
plot(hour, chance, main = "chance of getting a dinstinction")
```



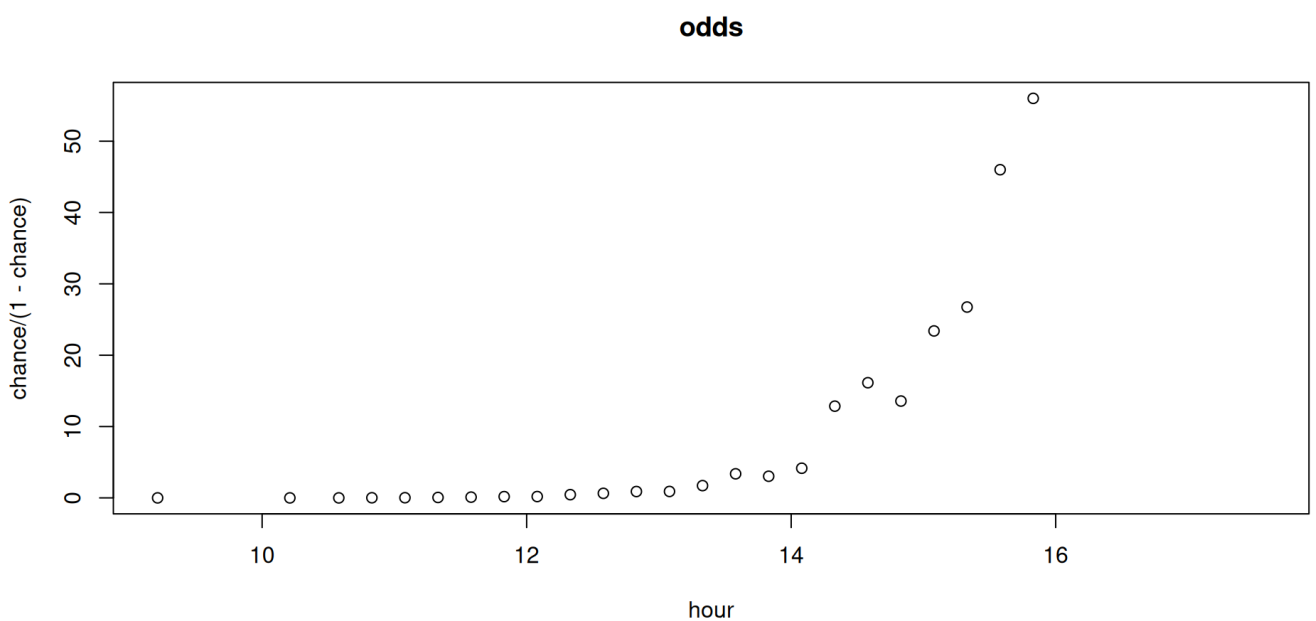
- How to build a model to fit this "chance" data?
    - ➠ Linear models don't work as chance is always bounded between 0 and 1.

Odds

- If an event is occurring with probability p, its odds is defined as

$$\text{odds} = \frac{\text{probability that event will occur}}{\text{probability that event will not occur}} = \frac{p}{1-p}$$

```
1 plot(hour, chance/(1 - chance), main = "odds")
```



- Above 0, but still too curvy to apply linear models

How to interpret odds

- Suppose that the probability that a student gets a distinction grade is 0.6 (i.e. $p = 0.6$).
  - Then the odds that a student gets a distinction grade is $\frac{0.6}{1-0.6} = 1.5$, which is 3 to 2.
  - This means the odds are 3 to 2 in favor of getting a distinction. That is, for every 2 students who don't get a distinction, 3 do.
- Interpretations of different odds values.
  - Odds > 1: The event is more likely to happen than not (favorable).
  - Odds = 1: The event is equally likely to happen or not (50–50 chance).
  - Odds < 1: The event is less likely to happen than not (unfavorable).
- Given the odds of an event, the corresponding probability of the event occuring is

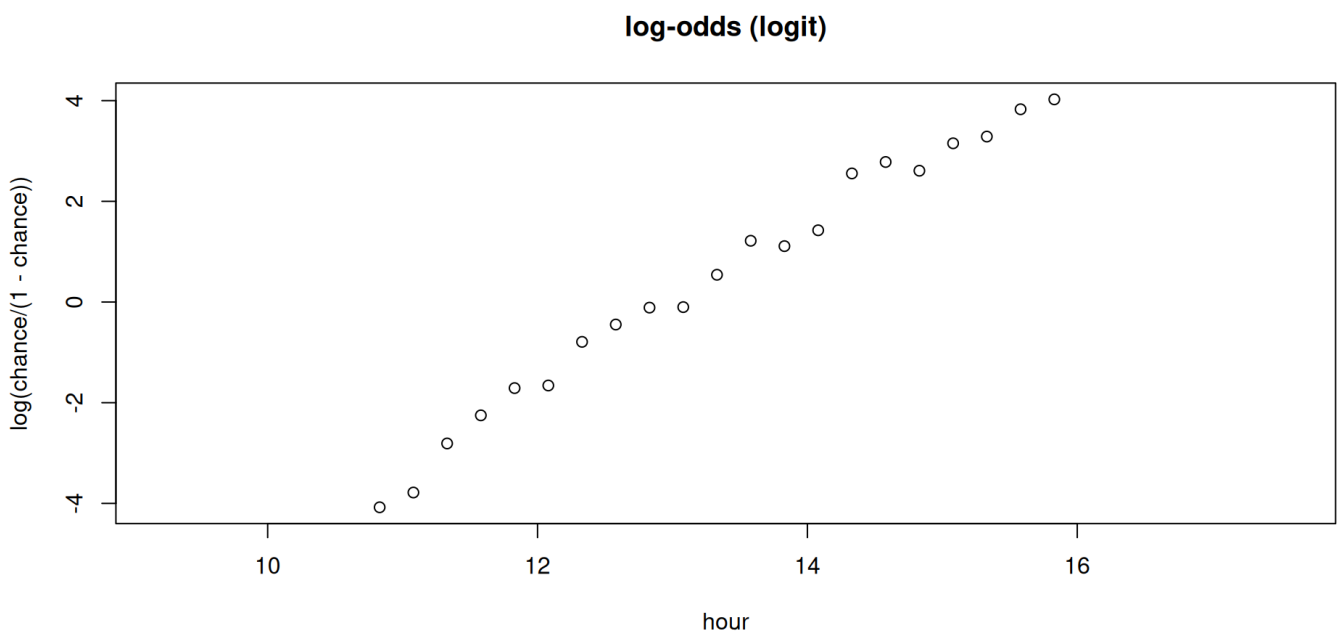
$$\mathrm{p} = \frac{\mathrm{odds}}{1 + \mathrm{odds}}$$

Log-odds (logit)

- We can use the log-odds (logit function) to transform the chance data

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

```
1 plot(hour, log(chance/(1 - chance)), main = "log-odds (logit)")
```



- Linear models have a chance to work well here

Logistic Regression

- Via the logit function, we can describe the relationship between a binary dependent variable Y and a set of independent variables $x_1, \ldots, x_k$.
- Suppose we have independent random draws $Y_1, \ldots, Y_n$,
  - where each Y_i is the sample sum of m_i independent random draws taken from a 0-1 box with success probability p_i (for getting 1's).
  - We often denote the model of the sample sum Y_i as

$$Y_i \sim \text{Binomial}(m_i, p_i)$$

- The success probability $\text{logit}(p_i)$ can be explained by varialbes $x_1, \ldots, x_k$ using the linear model

$$\text{logit}(p_i) = \log \frac{p_i}{1-p_i} = b_0 + b_1 \cdot x_{1,i} + b_2 \cdot x_{2,i} + \ldots + b_k \cdot x_{k,i}.$$

which also gives

$$Y_i \sim \text{Binomial}\left(m_i, \frac{\text{odds}_i}{1+\text{odds}_i}\right) \quad \text{where} \quad \text{odds}_i = \exp\left(\text{logit}(p_i)\right)$$

Fit a logistic regression model using `glm()`

```r
tab = cbind(distinction, total - distinction)
M1 = glm(tab ~ hour, family = "binomial")
summary(M1)
```

```
Call:
glm(formula = tab ~ hour, family = "binomial")

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -21.22639    0.77068  -27.54   <2e-16 ***
hour          1.63197    0.05895   27.68   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3693.884  on 24  degrees of freedom
Residual deviance:   26.703  on 23  degrees of freedom
AIC: 114.76

Number of Fisher Scoring iterations: 4
```

- `glm()` is the function used to perform generalized linear models.
- `family="binomial"` in glm fits a logistic regression.
- `tab = cbind(..., ...)` specifies both y_i and m_i for observing the binomial model.
  - ➠ the first column is number of successes (sample sum y_i) and the second column is number of failures $m_i - y_i$.

Interpret the fitted model

The fitted model is

$$\widehat{\text{logit}}(p) = -21.23 + 1.63 \times \text{hour}$$

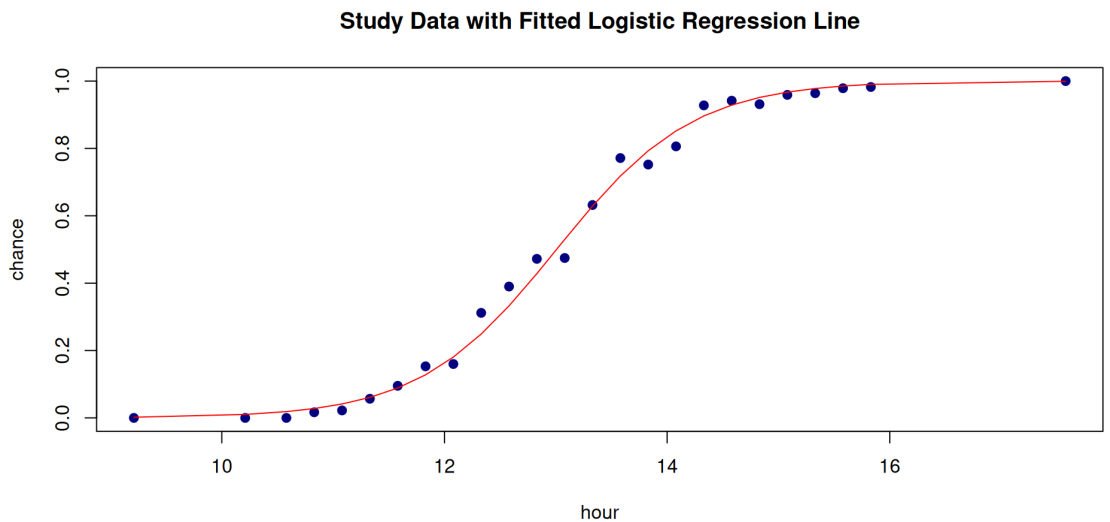
For each increase of one hour in study, the log-odds of achieving distinction increases by 1.63.

- Alternatively, the odds of achieving a distinction grade increase by $\exp(1.63) \approx 5.1$ times.
- Note that, the "change by $b_1 \times 100\%$" formula does not apply here, as it only applies to slopes that are much smaller than 1.

Access the fitted model

- `M1$fitted` gives the fitted chance

```
plot(hour, chance, pch = 19, col = "navy")
lines(hour, M1$fitted, type = "l", col = "red")
title(main = "Study Data with Fitted Logistic Regression Line")
```



- Model prediction: estimate the probability of achieving a distinction grade for a student who studies 15 hours per week.

```
predict(M1, data.frame(hour = 15), type = "response")
```

```
        1
0.9627854
```

Inference

```r
summary(M1)
```

```
Call:
glm(formula = tab ~ hour, family = "binomial")

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -21.22639    0.77068  -27.54   <2e-16 ***
hour          1.63197    0.05895   27.68   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3693.884  on 24  degrees of freedom
Residual deviance:   26.703  on 23  degrees of freedom
AIC: 114.76

Number of Fisher Scoring iterations: 4
```

- `glm()` uses the Z-test to assess the significance of an independent variable. The test statistic is $Z = \frac{\hat{b}_j}{SE(\hat{b}_j)} \sim N(0,1)$, where $SE(\hat{b}_j)$ is given in the output of `summary(glm(...))`.
- $H_0 : \beta_j = 0$ (the independent variable has no effect in explaining Y) and $H_1 : \beta_j \neq 0$ (the variable has an effect) are the default hypotheses.
- For multiple independent variables, we assess the effect of each variable after adjusting for all the others.

Additional notes

In the output of `summary(glm( ... ))`

- `glm()` uses a method called maximum likelihood estimate (MLE) to fit the logistic regression model.
  - ➟ If you try to directly fit a linear model to $\mathbf{logit}(\widehat{p_i})$), you may get a different result.
  - ➟ The maximum likelihood estimate is beyond the scope of this unit, so we only need to know how to apply the `glm()` function.
- Similarly, the derivation of $SE(\hat{b}_j)$ is also based on the MLE, so we skipped it.
- `deviance` is used to measure the quality of the model fit, **lower deviance means better fit.**
  - ➟ `Null deviance` is the he deviance of the null model including only the intercept (baseline prediction).
  - ➟ `Residual deviance` is the deviance of the alternative model with all specified independent variables.
  - ➟ We skip the derivations of the deviance (it's based on the concept of likelihood and maximum likelihood estimate)
- The **expected learning outcomes** are the formulation of the logistic regression model and how to use the output of `summary(glm( ... ))` to interpret the fitted model.

Case study of logistic regression

Pima Diabetes Database

The National Institute of Diabetes and Digestive and Kidney Diseases collected data from 724 female patients of Pima heritage. The dataset consists of several medical explanatory (independent) variables and one target (dependent) variable.

```r
df = read.csv("data/PimaCleaned.csv", header = T)
dim(df)
```

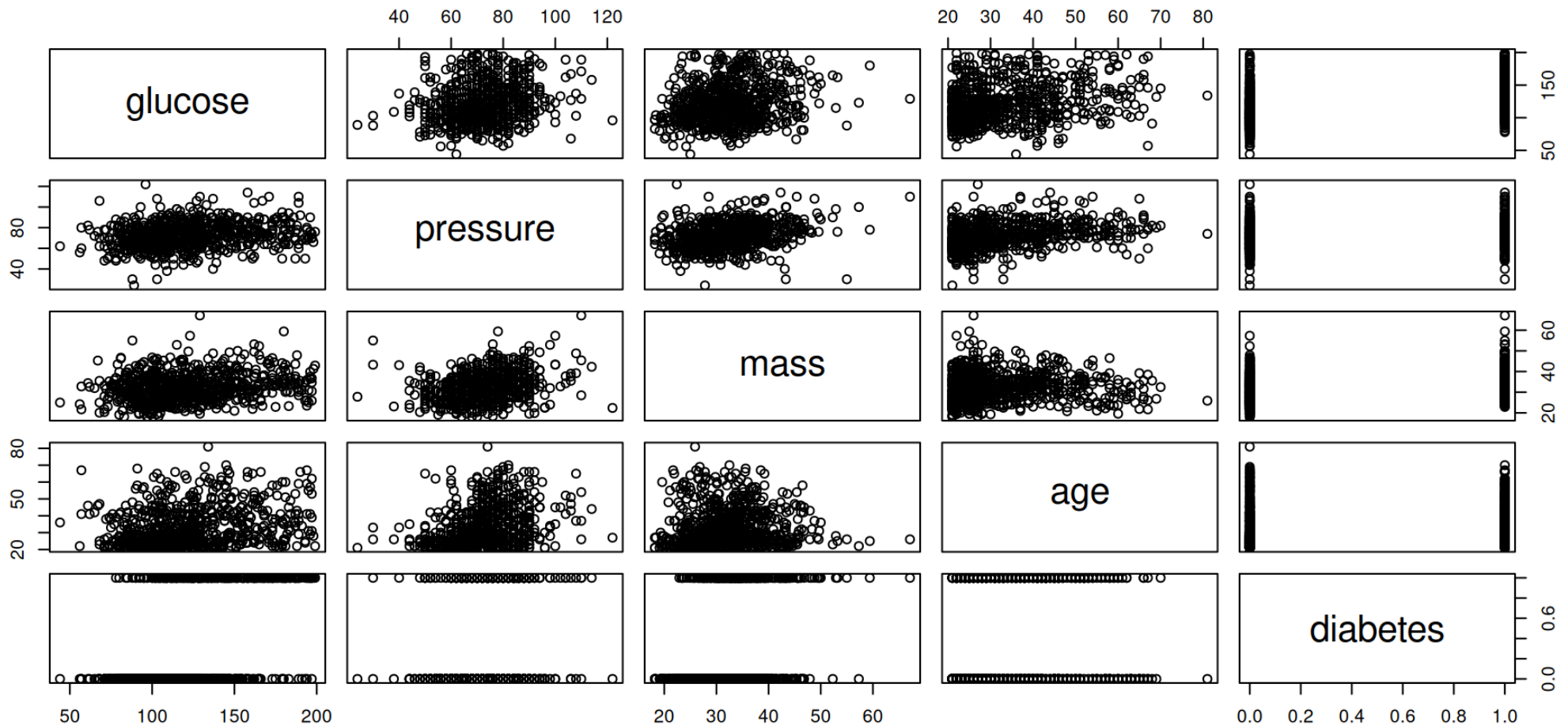
```
[1] 724   5
```

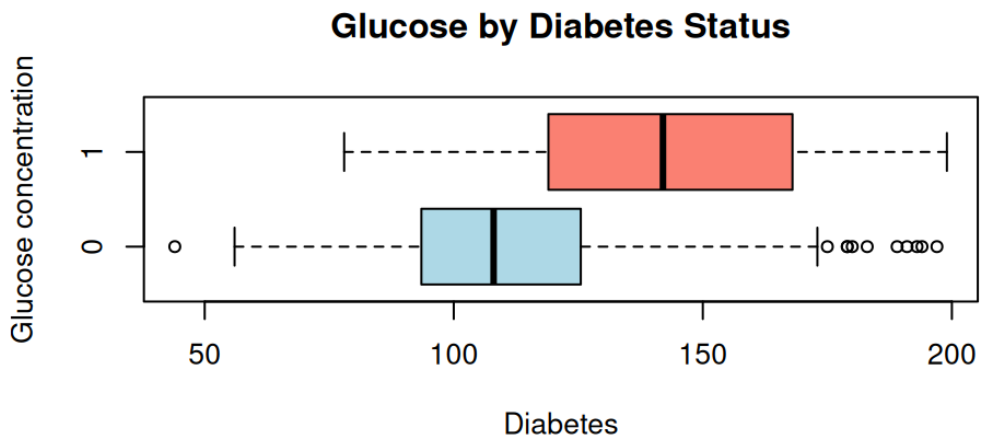
```r
str(df)
```

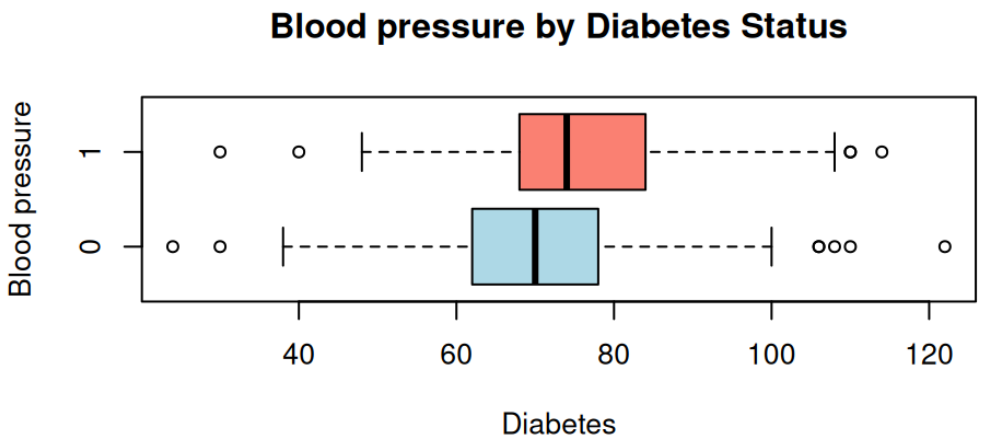
```
'data.frame':   724 obs. of  5 variables:
 $ glucose : int  148 85 183 89 137 116 78 197 110 168 ...
 $ pressure: int  72 66 64 66 40 74 50 70 92 74 ...
 $ mass    : num  33.6 26.6 23.3 28.1 43.1 25.6 31 30.5 37.6 38 ...
 $ age     : int  50 31 32 21 33 30 26 53 30 34 ...
 $ diabetes: int  1 0 1 0 1 0 1 1 0 1 ...
```

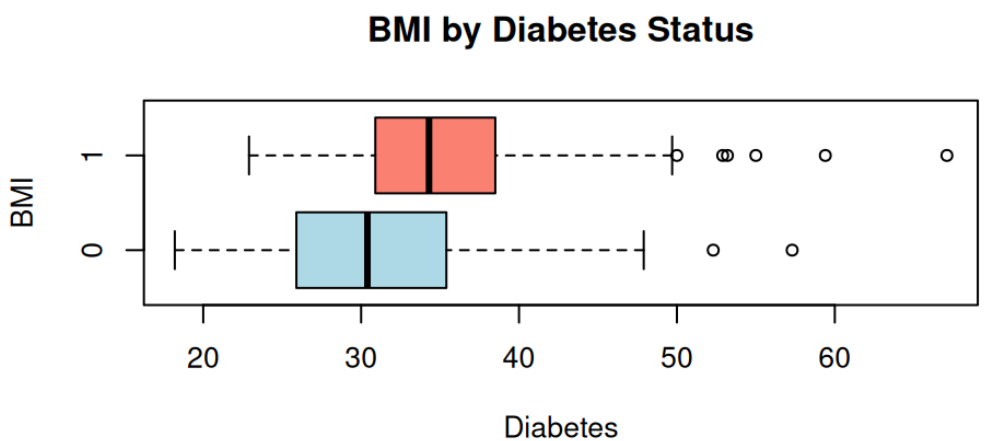
- `glucose`: plasma glucose concentration
- `pressure`: blood pressure (mm Hg)
- `mass`: body mass index
- `age`: age (years)
- `diabetes`: binary variable (`0` for negative and `1` for positive)

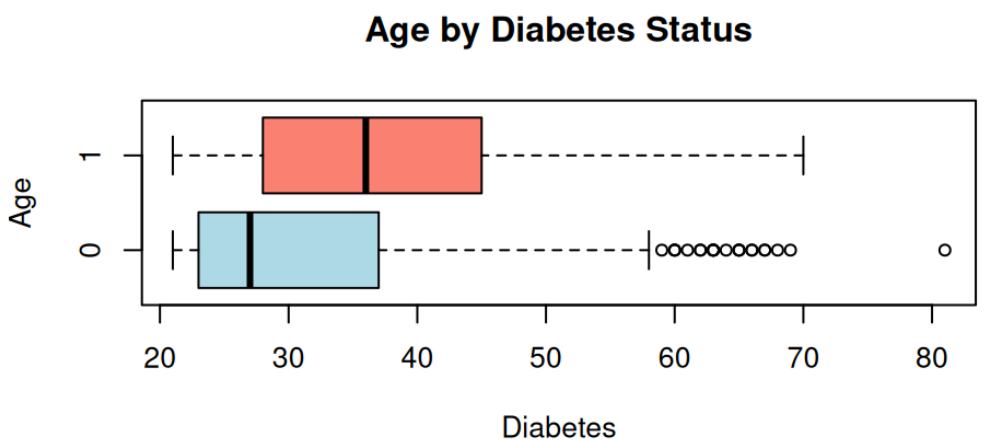
We want to predict diabetes status (the dependent varibale `diabetes`) using independent variables `glucose`, `pressure`, `mass`, and `age`.











It seems clear that there is some sort of a relationship between diabetes status and glucose concentration, perhaps even BMI, age, and blood pressure.

Fitted model

```r
model = glm(diabetes ~ ., data = df, family = binomial)
summary(model)$coefficients
```

```
               Estimate  Std. Error    z value     Pr(>|z|)
(Intercept) -8.673701182 0.795070256 -10.909352 1.039951e-27
glucose      0.034909453 0.003522771   9.909656 3.779463e-23
pressure    -0.008317124 0.008421586  -0.987596 3.233505e-01
mass         0.092529079 0.015314798   6.041809 1.523961e-09
age          0.034351783 0.008418056   4.080726 4.489521e-05
```

- If we're modelling 0-1 data (like in this case), we can directly use y_i as the dependent variable in `glm()`.
- If we're modelling grouped data (e.g., 3 out of 10 students got a distinction), we need to specify both the number of successes y_i and the total m_i
  - In the previous example, we pass `tab = cbind(distinction, total - distinction)` as the dependent variable to `glm()`
- The fitted model is

$$\widehat{\text{logit}}(p) = -8.674 + 0.0349 \cdot \text{glucose} - 0.0083 \cdot \text{pressure} + 0.0925 \cdot \text{mass} + 0.0344 \cdot \text{age}$$

- The interpretation and inference of logistic regression models with multiple variables are very similar to those of the multiple linear regression.

Interpretation

$$\widehat{\text{logit}}(p) = -8.674 + 0.0349 \cdot \text{glucose} - 0.0083 \cdot \text{pressure} + 0.0925 \cdot \text{mass} + 0.0344 \cdot \text{age}$$

Examples of possible interpretations:

- Holding other variables constant, a one-year increase in age is associated with a 0.0344 increase in the log-odds of having diabetes;
  - that is approximately 3.44% increase in the odds of having diabetes.
  - The "change by $b_1 \times 100\%$" formula applies to the slope here.
- Holding other variables constant, a one unit increase in glucose concentration is associated with an approximately 3.49% increase in the odds of having diabetes.

Inference

```r
round(summary(model)$coefficients, 6)
```

```
             Estimate Std. Error    z value Pr(>|z|)
(Intercept) -8.673701   0.795070 -10.909352 0.000000
glucose      0.034909   0.003523   9.909656 0.000000
pressure    -0.008317   0.008422  -0.987596 0.323351
mass         0.092529   0.015315   6.041809 0.000000
age          0.034352   0.008418   4.080726 0.000045
```

At the 5% level of significance, after adjusting for all other independent variables in the model:

- Glucose concentration has a significant effect on the chance of having diabetes.
- The data is consistent with the null hypothesis that blood pressure does not have an effect on the chance of having diabetes.