Medical Statistics Using R: Part 1

Short version. Draft updated August 13, 2018. Not for sale :-)

Wan Nor Arifin

Contents

Chapter 1

Linear Regression

1.1 Introduction

1. A statistical method to model relationship between:
   - outcome: numerical variable.
   - predictors/independent variables: numerical, categorical variables.
2. A type of Generalized Linear Models (GLMs), which also includes other outcome types, e.g. categorical and count.
3. Basically, the linear relationship is structured as follows,

$$numerical\ outcome = numerical\ predictors + categorical\ predictors$$

1.2 Simple linear regression (SLR)

About SLR

1. Model *linear* (straight line) relationship between:
   - outcome: numerical variable.
   - a predictor: numerical variable (only).

   Note: What if the predictor is a categorical variable? Remember, we already handled that with one-way ANOVA.
2. Formula,

   $$numerical\ outcome = intercept + coefficient \times numerical\ predictor$$

   in short,

   $$\hat{y} = \beta_0 + \beta_1 x_1$$

   where $\hat{y}$ is the predicted value of the outcome y.

Analysis

```
# library
library(foreign)
library(epiDisplay)
library(psych)
library(lattice)
library(rsq)
library(MASS)
library(car)
```

```
# data
coronary = read.dta("coronary.dta")
str(coronary)
```

```
## 'data.frame':    200 obs. of  9 variables:
##  $ id    : num  1 14 56 61 62 64 69 108 112 134 ...
##  $ cad   : Factor w/ 2 levels "no cad","cad": 1 1 1 1 1 1 2 1 1 1 ...
##  $ sbp   : num  106 130 136 138 115 124 110 112 138 104 ...
##  $ dbp   : num  68 78 84 100 85 72 80 70 85 70 ...
##  $ chol  : num  6.57 6.33 5.97 7.04 6.66 ...
##  $ age   : num  60 34 36 45 53 43 44 50 43 48 ...
##  $ bmi   : num  38.9 37.8 40.5 37.6 40.3 ...
##  $ race  : Factor w/ 3 levels "malay","chinese",..: 3 1 1 1 3 1 1 2 2 2 ...
##  $ gender: Factor w/ 2 levels "woman","man": 1 1 1 1 2 2 2 1 1 2 ...
##  - attr(*, "datalabel")= chr "Written by R.              "
##  - attr(*, "time.stamp")= chr ""
##  - attr(*, "formats")= chr  "%9.0g" "%9.0g" "%9.0g" "%9.0g" ...
##  - attr(*, "types")= int  100 108 100 100 100 100 100 108 108
##  - attr(*, "val.labels")= chr  "" "cad" "" "" ...
##  - attr(*, "var.labels")= chr  "id" "cad" "sbp" "dbp" ...
##  - attr(*, "version")= int 7
##  - attr(*, "label.table")=List of 3
##   ..$ cad   : Named int  1 2
##   .. ..- attr(*, "names")= chr  "no cad" "cad"
##   ..$ race  : Named int  1 2 3
##   .. ..- attr(*, "names")= chr  "malay" "chinese" "indian"
##   ..$ gender: Named int  1 2
##   .. ..- attr(*, "names")= chr  "woman" "man"
```

1.2.1 Data exploration

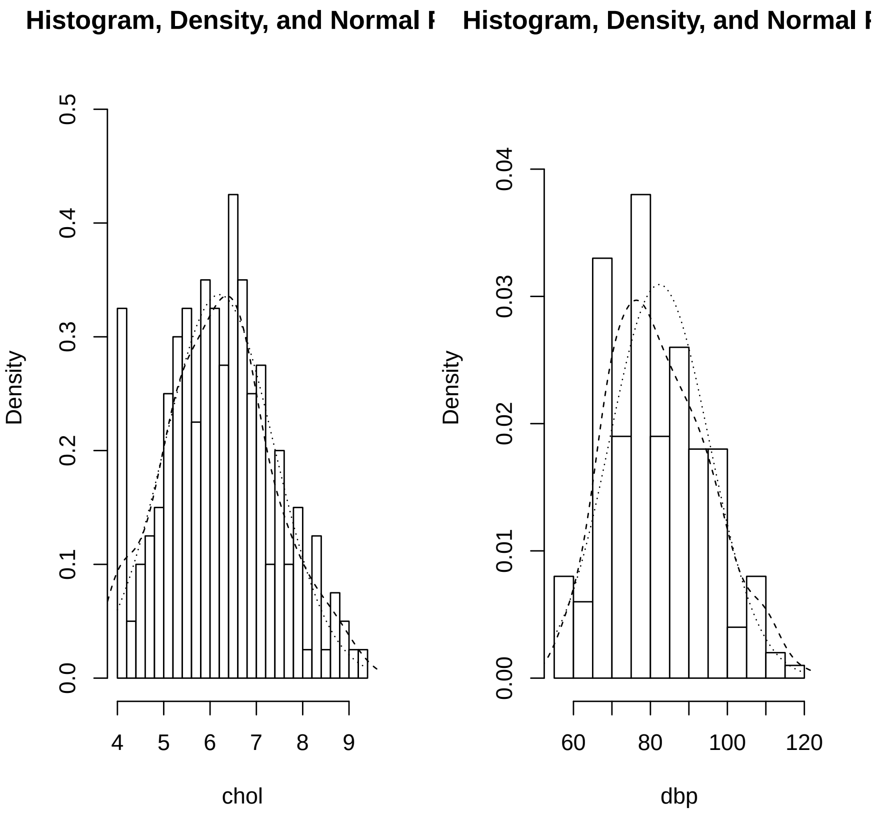
1.2.1.1 Descriptive statistics

```
summ(coronary[c("chol", "dbp")])
```

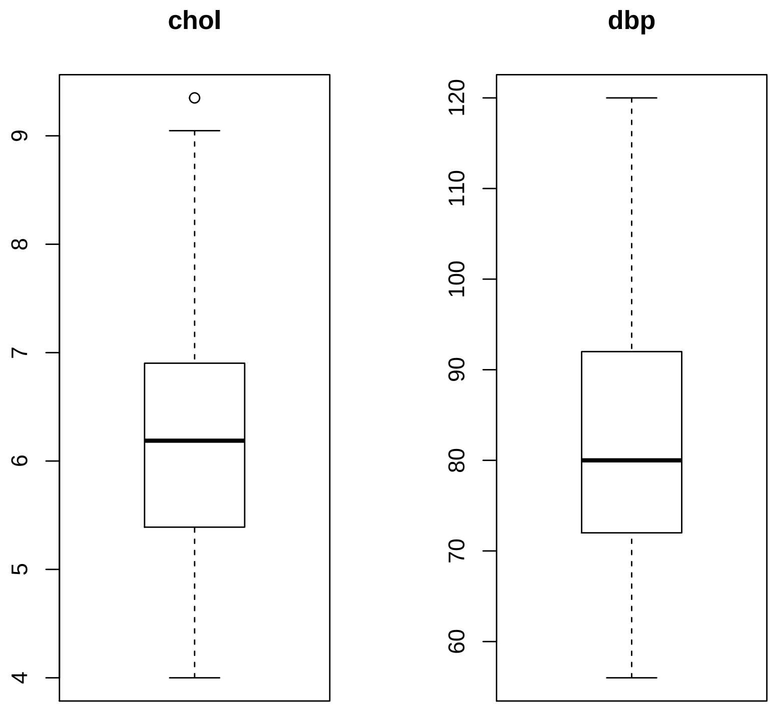
```
## 
## No. of observations = 200
## 
##   Var. name obs. mean   median  s.d.   min.   max.
## 1 chol      200  6.2    6.19    1.18   4      9.35
## 2 dbp       200  82.31  80      12.9   56     120
```

1.2.1.2 Plots

```r
multi.hist(coronary[c("chol", "dbp")], ncol = 2)
```



```r
par(mfrow = c(1, 2))
mapply(boxplot, coronary[c("chol", "dbp")],
       main = colnames(coronary[c("chol", "dbp")]))
```



```
##       chol      dbp
## stats Numeric,5 Numeric,5
## n     200       200
## conf  Numeric,2 Numeric,2
## out   9.35      Numeric,0
## group 1         Numeric,0
## names ""        ""
```

```
par(mfrow = c(1, 1))
```

1.2.2 Univariable

Fit model,

```
# model: chol ~ dbp
slr_chol = glm(chol ~ dbp, data = coronary)
summary(slr_chol)
```

```
##
## Call:
## glm(formula = chol ~ dbp, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9967  -0.8304  -0.1292   0.7734   2.8470
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.995134   0.492092   6.087 5.88e-09 ***
## dbp         0.038919   0.005907   6.589 3.92e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.154763)
##
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 228.64  on 198  degrees of freedom
## AIC: 600.34
##
## Number of Fisher Scoring iterations: 2
```

```
Confint(slr_chol)  # 95% CI
```

```
##               Estimate      2.5 %     97.5 %
## (Intercept) 2.99513427 2.03065127 3.95961727
## dbp         0.03891876 0.02734161 0.05049591
```

Important results,

- Coefficient, β.
- 95% CI.
- P-value.

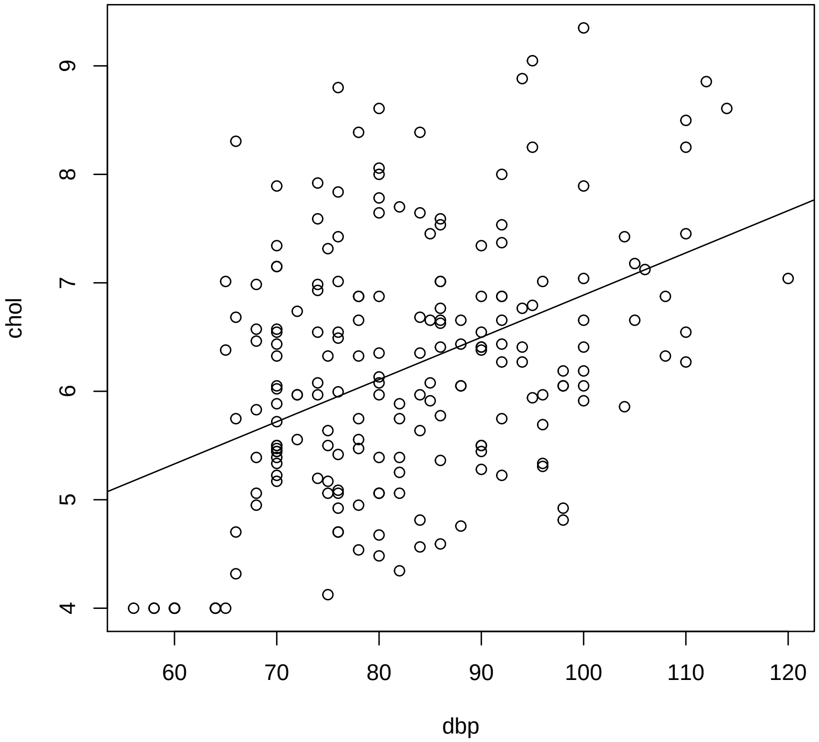
Obtain R^2, % of variance explained,

```
rsq(slr_chol, adj = T)
```

```
## [1] 0.1756834
```

Scatter plot,

```
plot(chol ~ dbp, data = coronary)
abline(slr_chol)
```



this allows assessment of normality, linearity and equal variance assumptions. We expect eliptical/oval shape (normality), equal scatter of dots on both sides of the prediction line (equal variance). Both these indicate linear relationship between `chol` and `dbp`.

1.2.3 Interpretation

- 1mmHg increase in DBP causes 0.04mmol/L increase in cholestrol.
- DBP explains 17.6% variance in cholestrol.

1.2.4 Model equation

$$chol = 3.0 + 0.04 \times dbp$$

1.3 Multiple linear regression (MLR)

About MLR

1. Model *linear* relationship between:
   - outcome: numerical variable.
   - predictors: numerical, categorical variables.

Note: MLR is a term that refers to linear regression with two or more *numerical* variables. Whenever we have both numerical and categorical variables, the proper term for the regression model is *General Linear Model*. However, we will use the term MLR in this workshop.

2. Formula,

$$\begin{aligned} numerical\ outcome = {} & intercept + coefficients \times numerical\ predictors \\ & + coefficients \times categorical\ predictors \end{aligned}$$

in a shorter form,

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k$$

where we have k predictors.

Whenever the predictor is a categorical variable with more than two levels, we use dummy variable(s). This can be easily specified in R using `factor()` if the variable is not yet properly specified as such. There is no problem with binary categorical variable.

For a categorical variable with more than two levels, the number of dummy variables (i.e. once turned into several binary variables) equals number of levels minus one. For example, whenever we have four levels, we will obtain three dummy (binary) variables.

Analysis

```r
# data
str(coronary)
```

```
## 'data.frame':    200 obs. of  9 variables:
##  $ id    : num  1 14 56 61 62 64 69 108 112 134 ...
##  $ cad   : Factor w/ 2 levels "no cad","cad": 1 1 1 1 1 1 2 1 1 1 ...
##  $ sbp   : num  106 130 136 138 115 124 110 112 138 104 ...
##  $ dbp   : num  68 78 84 100 85 72 80 70 85 70 ...
##  $ chol  : num  6.57 6.33 5.97 7.04 6.66 ...
##  $ age   : num  60 34 36 45 53 43 44 50 43 48 ...
##  $ bmi   : num  38.9 37.8 40.5 37.6 40.3 ...
##  $ race  : Factor w/ 3 levels "malay","chinese",..: 3 1 1 1 3 1 1 2 2 2 ...
##  $ gender: Factor w/ 2 levels "woman","man": 1 1 1 1 2 2 2 1 1 2 ...
##  - attr(*, "datalabel")= chr "Written by R.              "
##  - attr(*, "time.stamp")= chr ""
##  - attr(*, "formats")= chr  "%9.0g" "%9.0g" "%9.0g" "%9.0g" ...
##  - attr(*, "types")= int  100 108 100 100 100 100 100 108 108
##  - attr(*, "val.labels")= chr  "" "cad" "" "" ...
##  - attr(*, "var.labels")= chr  "id" "cad" "sbp" "dbp" ...
##  - attr(*, "version")= int 7
##  - attr(*, "label.table")=List of 3
##   ..$ cad   : Named int  1 2
##   .. ..- attr(*, "names")= chr  "no cad" "cad"
##   ..$ race  : Named int  1 2 3
##   .. ..- attr(*, "names")= chr  "malay" "chinese" "indian"
##   ..$ gender: Named int  1 2
##   .. ..- attr(*, "names")= chr  "woman" "man"
```

We exclude `id`, `cad` and `age` from our data for the purpose of this analysis, keeping only `sbp` , `dbp`, `bmi`, `race` and `gender`. We will add `age` later in the exercise.

```r
coronary = subset(coronary, select = -c(id, cad, age))
# remove id, cad, age from our data since we're not going to use them,
# easier to specifiy multivariable model.
```

1.3.1 Data exploration

1.3.1.1 Descriptive statistics

```
summ(coronary[c("chol", "sbp", "dbp", "bmi")])
```

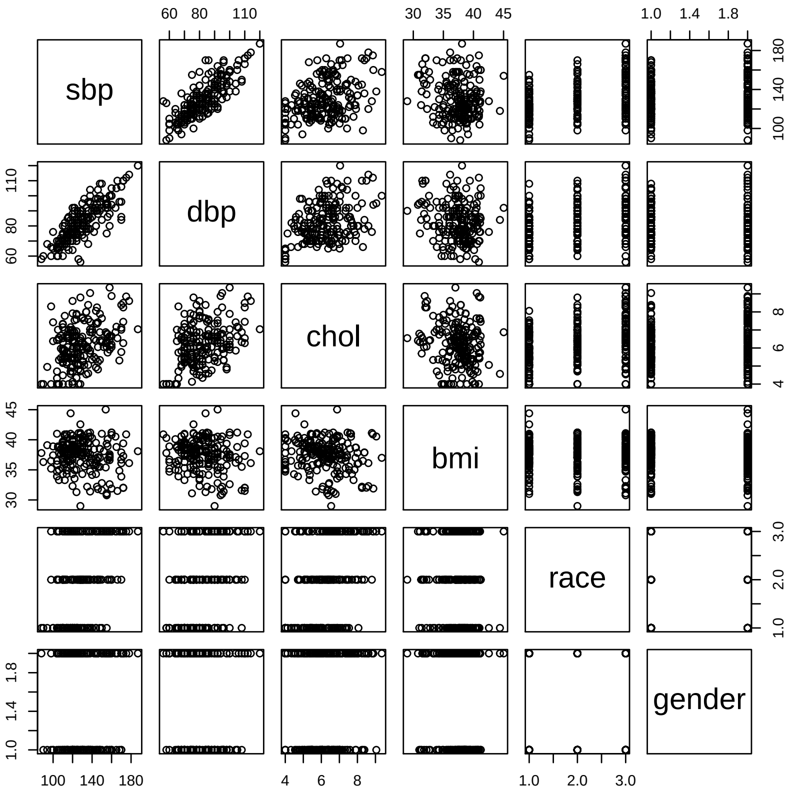
```
##
## No. of observations = 200
##
##   Var. name obs. mean   median  s.d.   min.   max.
## 1 chol      200  6.2    6.19    1.18   4      9.35
## 2 sbp       200  130.18 126     19.81  88     187
## 3 dbp       200  82.31  80      12.9   56     120
## 4 bmi       200  37.45  37.8    2.68   28.99  45.03
```

```
codebook(coronary[c("race", "gender")])
```

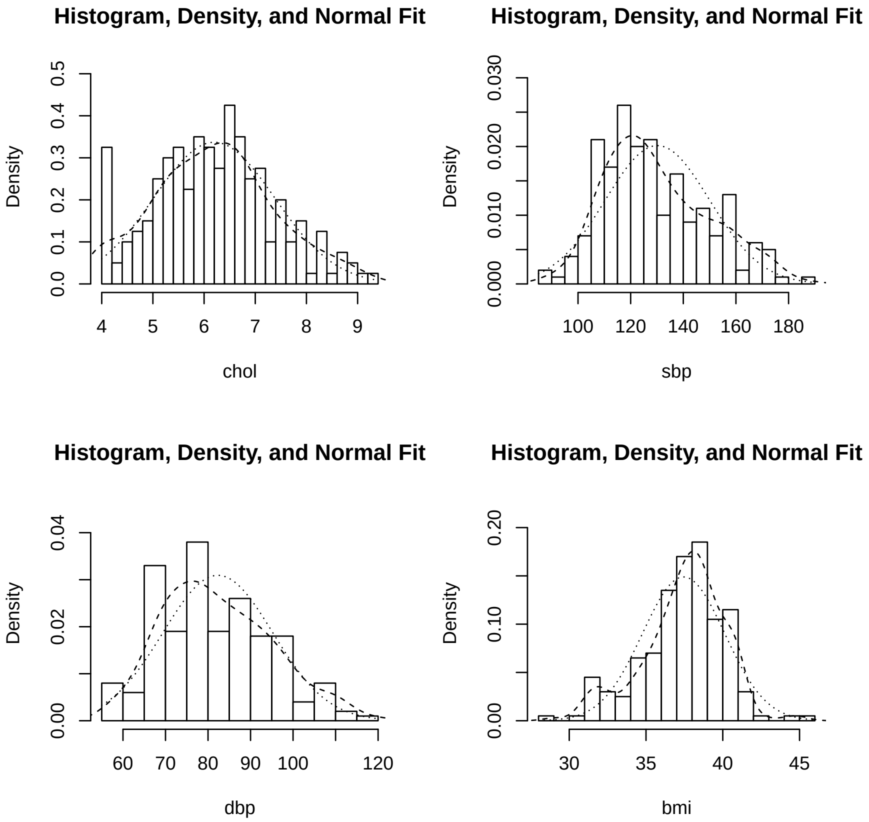
```
##
##
##
## race      :
##         Frequency Percent
## malay          73    36.5
## chinese        64    32.0
## indian         63    31.5
##
##  ==================
## gender    :
##       Frequency Percent
## woman       100      50
## man         100      50
##
##  ==================
```

1.3.1.2 Plots

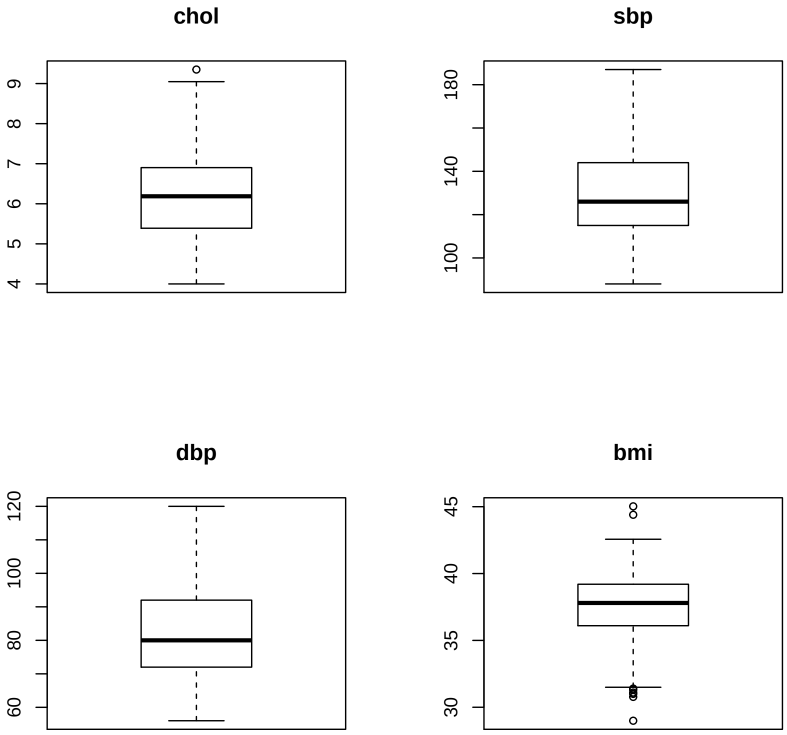
```
plot(coronary)
```



```
multi.hist(coronary[c("chol", "sbp", "dbp", "bmi")])
```

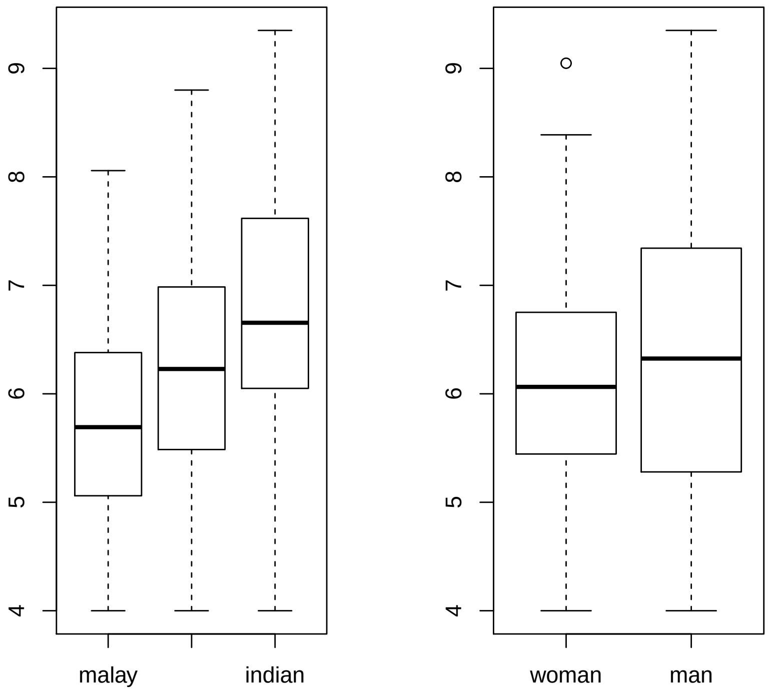


```r
par(mfrow = c(2, 2))
mapply(boxplot, coronary[c("chol", "sbp", "dbp", "bmi")],
       main = colnames(coronary[c("chol", "sbp", "dbp", "bmi")]))
```



```
##       chol      sbp       dbp       bmi
## stats Numeric,5 Numeric,5 Numeric,5 Numeric,5
## n     200       200       200       200
## conf  Numeric,2 Numeric,2 Numeric,2 Numeric,2
## out   9.35      Numeric,0 Numeric,0 Numeric,8
## group 1         Numeric,0 Numeric,0 Numeric,8
## names ""        ""        ""        ""
```

```
par(mfrow = c(1, 1))
par(mfrow = c(1, 2))
boxplot(chol ~ race, data = coronary)
boxplot(chol ~ gender, data = coronary)
```



```r
par(mfrow = c(1, 1))
```

1.3.2 Variable selection

1.3.2.1 Univariable

Perform SLR for `chol, sbp, dbp` and `bmi` on your own as shown above. Now, we are concerned with which variables are worthwhile to include in the multivariable models.

We want to choose only variables with P-values < 0.25 to be included in MLR. Obtaining the P-values for each variable is easy by LR test,

```r
slr_chol0 = glm(chol ~ 1, data = coronary)
summary(slr_chol0)
```

```
##
## Call:
## glm(formula = chol ~ 1, data = coronary)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.19854  -0.80854  -0.01104   0.69021   3.15146
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.19854    0.08369   74.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.400874)
##
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 278.77  on 199  degrees of freedom
## AIC: 637.99
##
## Number of Fisher Scoring iterations: 2
```

```
names(coronary)
```

```
## [1] "sbp"    "dbp"    "chol"   "bmi"    "race"   "gender"
```

```
add1(slr_chol0, scope = ~ sbp + dbp + bmi + race + gender, test = "LRT")
```

```
## Single term additions
##
## Model:
## chol ~ 1
##        Df Deviance    AIC scaled dev.  Pr(>Chi)
## <none>      278.77 637.99
## sbp     1   235.36 606.14      33.855 5.938e-09 ***
## dbp     1   228.64 600.34      39.648 3.042e-10 ***
## bmi     1   272.17 635.20       4.792   0.02859 *
## race    2   241.68 613.43      28.561 6.280e-07 ***
## gender  1   277.45 639.04       0.952   0.32933
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All variables are significant and $< .25$ except `gender`. So proceed with the rest of the variables, excluding `gender`.

1.3.2.2 Multivariable

Perform MLR with *all* selected variables,

```
# all
mlr_chol = glm(chol ~ sbp + dbp + bmi + race, data = coronary)
#mlr_chol = glm(chol ~ ., data = coronary)  # shortcut
summary(mlr_chol)
```

```
##
## Call:
## glm(formula = chol ~ sbp + dbp + bmi + race, data = coronary)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.17751  -0.73860  -0.02674   0.63163   2.90926
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.842338   1.265149   3.827 0.000175 ***
## sbp          0.000975   0.006990   0.139 0.889210
## dbp          0.028350   0.010327   2.745 0.006615 **
## bmi         -0.038537   0.028170  -1.368 0.172879
## racechinese  0.354039   0.183169   1.933 0.054710 .
## raceindian   0.716327   0.200346   3.575 0.000441 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.089387)
##
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 211.34  on 194  degrees of freedom
## AIC: 592.61
##
## Number of Fisher Scoring iterations: 2
```

```
rsq(mlr_chol, adj = T)
```

```
## [1] 0.2223518
```

Focus on,

- Coefficients, βs.
- 95% CI.
- P-values.

For model fit,

- R^2 – % of variance explained by the model.
- Akaike Information Criterion, AIC – for comparison with other models. This is not useful alone, but for comparison with other models. The model with the lowest AIC is the best model.

1.3.2.3 Stepwise

As you can see, not all variables are significant. How to select? We proceed with stepwise automatic selection,

```
# stepwise
# both
mlr_chol_stepboth = step(mlr_chol, direction = "both")
```

```
## Start:  AIC=592.61
## chol ~ sbp + dbp + bmi + race
##
##        Df Deviance    AIC
## - sbp   1   211.36 590.63
## - bmi   1   213.38 592.53
## <none>      211.34 592.61
## - dbp   1   219.55 598.23
## - race  2   225.30 601.40
##
## Step:  AIC=590.63
## chol ~ dbp + bmi + race
##
##        Df Deviance    AIC
## - bmi   1   213.40 590.55
## <none>      211.36 590.63
## + sbp   1   211.34 592.61
## - race  2   227.04 600.94
## - dbp   1   235.88 610.58
```

```
##
## Step:  AIC=590.55
## chol ~ dbp + race
##
##         Df Deviance    AIC
## <none>       213.40 590.55
## + bmi    1   211.36 590.63
## + sbp    1   213.38 592.53
## - race   2   228.64 600.34
## - dbp    1   241.68 613.43
```

```
summary(mlr_chol_stepboth)  # racechinese marginally sig.
```

```
##
## Call:
## glm(formula = chol ~ dbp + race, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1378  -0.7068  -0.0289   0.5997   2.7778
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.298028   0.486213   6.783 1.36e-10 ***
## dbp         0.031108   0.006104   5.096 8.14e-07 ***
## racechinese 0.359964   0.182149   1.976 0.049534 *
## raceindian  0.713690   0.190883   3.739 0.000243 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.088777)
##
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 213.40  on 196  degrees of freedom
## AIC: 590.55
##
## Number of Fisher Scoring iterations: 2
```

```
# forward
mlr_chol_stepforward = step(slr_chol0, scope = ~ sbp + dbp + bmi + race + gender,
                            direction = "forward")
```

```
## Start:  AIC=637.99
## chol ~ 1
##
##           Df Deviance    AIC
## + dbp      1   228.64 600.34
## + sbp      1   235.36 606.14
## + race     2   241.68 613.43
## + bmi      1   272.17 635.20
## <none>         278.77 637.99
## + gender   1   277.45 639.04
##
## Step:  AIC=600.34
## chol ~ dbp
##
```

```
##          Df Deviance    AIC
## + race    2   213.40 590.55
## <none>        228.64 600.34
## + gender  1   226.64 600.58
## + sbp     1   226.96 600.87
## + bmi     1   227.04 600.94
##
## Step:  AIC=590.55
## chol ~ dbp + race
##
##          Df Deviance    AIC
## <none>        213.40 590.55
## + bmi     1   211.36 590.63
## + gender  1   212.47 591.67
## + sbp     1   213.38 592.53
```

```
summary(mlr_chol_stepforward)  # same with both
```

```
##
## Call:
## glm(formula = chol ~ dbp + race, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1378  -0.7068  -0.0289   0.5997   2.7778
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.298028   0.486213   6.783 1.36e-10 ***
## dbp         0.031108   0.006104   5.096 8.14e-07 ***
## racechinese 0.359964   0.182149   1.976 0.049534 *
## raceindian  0.713690   0.190883   3.739 0.000243 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.088777)
##
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 213.40  on 196  degrees of freedom
## AIC: 590.55
##
## Number of Fisher Scoring iterations: 2
```

```
# backward
mlr_chol_stepback = step(mlr_chol, direction = "backward")
```

```
## Start:  AIC=592.61
## chol ~ sbp + dbp + bmi + race
##
##        Df Deviance    AIC
## - sbp   1   211.36 590.63
## - bmi   1   213.38 592.53
## <none>      211.34 592.61
## - dbp   1   219.55 598.23
## - race  2   225.30 601.40
```

```
##
## Step:  AIC=590.63
## chol ~ dbp + bmi + race
##
##          Df Deviance    AIC
## - bmi     1   213.40 590.55
## <none>        211.36 590.63
## - race    2   227.04 600.94
## - dbp     1   235.88 610.58
##
## Step:  AIC=590.55
## chol ~ dbp + race
##
##          Df Deviance    AIC
## <none>        213.40 590.55
## - race    2   228.64 600.34
## - dbp     1   241.68 613.43
```

```
summary(mlr_chol_stepback)  # same with both
```

```
##
## Call:
## glm(formula = chol ~ dbp + race, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1378  -0.7068  -0.0289   0.5997   2.7778
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.298028   0.486213   6.783 1.36e-10 ***
## dbp         0.031108   0.006104   5.096 8.14e-07 ***
## racechinese 0.359964   0.182149   1.976 0.049534 *
## raceindian  0.713690   0.190883   3.739 0.000243 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.088777)
##
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 213.40  on 196  degrees of freedom
## AIC: 590.55
##
## Number of Fisher Scoring iterations: 2
```

Looking at all these results, we choose:

```
chol ~ dbp + race
```

which has the lowest AIC.

```
mlr_chol1 = glm(chol ~ dbp + race, data = coronary)
summary(mlr_chol1)
```

```
##
## Call:
## glm(formula = chol ~ dbp + race, data = coronary)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1378  -0.7068  -0.0289   0.5997   2.7778
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.298028   0.486213   6.783 1.36e-10 ***
## dbp         0.031108   0.006104   5.096 8.14e-07 ***
## racechinese 0.359964   0.182149   1.976 0.049534 *
## raceindian  0.713690   0.190883   3.739 0.000243 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.088777)
##
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 213.40  on 196  degrees of freedom
## AIC: 590.55
##
## Number of Fisher Scoring iterations: 2
```

1.3.2.4 Confounder

If we include a variable and it causes notable change ($> 20\%$) in the coefficients of other variables, it is a confounder. When the confounder is significant and the main effect variable is also significant, we keep the confounder in the model.

Formula for % change,

> ```
> 100 * (model_small - model_large) / model_large
> ```
>
> Hosmer, Lemeshow, & Sturdivant (2013)

Start by including common demographic adjustment, gender,

```r
# + gender
mlr_chol2 = glm(chol ~ dbp + race + gender, data = coronary)
summary(mlr_chol2)  # higher AIC, gender insig.
```

```
##
## Call:
## glm(formula = chol ~ dbp + race + gender, data = coronary)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.06350  -0.71634  -0.04471   0.64533   2.70974
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.203032   0.497111   6.443 8.94e-10 ***
## dbp         0.031533   0.006124   5.149 6.37e-07 ***
## racechinese 0.353052   0.182369   1.936   0.0543 .
## raceindian  0.692724   0.192293   3.602   0.0004 ***
## genderman   0.137663   0.148790   0.925   0.3560
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.089578)
##
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 212.47  on 195  degrees of freedom
## AIC: 591.67
##
## Number of Fisher Scoring iterations: 2
```

```
coef(mlr_chol2); coef(mlr_chol1)
```

```
## (Intercept)         dbp racechinese  raceindian   genderman
##   3.2030318   0.0315331   0.3530516   0.6927239   0.1376627

## (Intercept)         dbp racechinese  raceindian
##  3.29802826  0.03110811  0.35996365  0.71369024
```

```
100 * (coef(mlr_chol1) - coef(mlr_chol2)[1:4])/coef(mlr_chol2)[1:4]  # change < 20%
```

```
## (Intercept)         dbp racechinese  raceindian
##    2.965828   -1.347773    1.957792    3.026647
```

```
# no notable change in coeffs, gender is not a confounder
```

Now, we can try adding sbp & bmi to mlr_chol1 and see what happens to the coefficients. We will use update() function here.

```
mlr_chol3 = update(mlr_chol1, . ~ . + sbp)
summary(mlr_chol3)  # higher AIC, sbp insig.
```

```
##
## Call:
## glm(formula = chol ~ dbp + race + sbp, data = coronary)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.12850  -0.71572  -0.03242   0.59676   2.77189
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.269724   0.529556   6.174 3.78e-09 ***
## dbp         0.029978   0.010281   2.916 0.003963 **
## racechinese 0.357407   0.183561   1.947 0.052963 .
## raceindian  0.705445   0.200635   3.516 0.000545 ***
## sbp         0.000958   0.007005   0.137 0.891365
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.094256)
##
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 213.38  on 195  degrees of freedom
## AIC: 592.53
##
## Number of Fisher Scoring iterations: 2
```

```
coef(mlr_chol3); coef(mlr_chol1)
```

```
##  (Intercept)          dbp  racechinese   raceindian          sbp 
## 3.2697237312 0.0299783153 0.3574065705 0.7054452332 0.0009580065

## (Intercept)         dbp racechinese  raceindian 
##  3.29802826  0.03110811  0.35996365  0.71369024
```

```
100 * (coef(mlr_chol1) - coef(mlr_chol3)[1:4])/coef(mlr_chol3)[1:4]  # change < 20%
```

```
## (Intercept)         dbp racechinese  raceindian 
##   0.8656550   3.7687027   0.7154536   1.1687670
```

```
# no notable change in coeffs, sbp is not a confounder
```

```
mlr_chol4 = update(mlr_chol1, . ~ . + bmi)
summary(mlr_chol4)  # slighly higher AIC, bmi insig.
```

```
## 
## Call:
## glm(formula = chol ~ dbp + race + bmi, data = coronary)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -2.18698  -0.73076  -0.01935   0.63476   2.91524  
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  4.870859   1.245373   3.911 0.000127 ***
## dbp          0.029500   0.006203   4.756 3.83e-06 ***
## racechinese  0.356642   0.181757   1.962 0.051164 .  
## raceindian   0.724716   0.190625   3.802 0.000192 ***
## bmi         -0.038530   0.028099  -1.371 0.171871    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 1.083909)
## 
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 211.36  on 195  degrees of freedom
## AIC: 590.63
## 
## Number of Fisher Scoring iterations: 2
```

```
coef(mlr_chol4); coef(mlr_chol1)
```

```
## (Intercept)         dbp racechinese  raceindian         bmi 
##  4.87085865  0.02950027  0.35664168  0.72471631 -0.03853042

## (Intercept)         dbp racechinese  raceindian 
##  3.29802826  0.03110811  0.35996365  0.71369024
```

```
100 * (coef(mlr_chol1) - coef(mlr_chol4)[1:4])/coef(mlr_chol4)[1:4]  # change < 20%
```

```
## (Intercept)         dbp racechinese  raceindian 
##  -32.290619    5.450250    0.931459   -1.521432
```

```
# no notable change in coeffs of other vars (ignore intercept!)
# bmi is not a confounder
```

Our chosen model:

```
    mlr_chol1: chol ~ dbp + race
```

```
summary(mlr_chol1)
```

```
##
## Call:
## glm(formula = chol ~ dbp + race, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1378  -0.7068  -0.0289   0.5997   2.7778
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.298028   0.486213   6.783 1.36e-10 ***
## dbp         0.031108   0.006104   5.096 8.14e-07 ***
## racechinese 0.359964   0.182149   1.976 0.049534 *
## raceindian  0.713690   0.190883   3.739 0.000243 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.088777)
##
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 213.40  on 196  degrees of freedom
## AIC: 590.55
##
## Number of Fisher Scoring iterations: 2
```

```
Confint(mlr_chol1)  # 95% CI of the coefficients
```

```
##               Estimate       2.5 %     97.5 %
## (Intercept) 3.29802826 2.345067995 4.25098852
## dbp         0.03110811 0.019143668 0.04307255
## racechinese 0.35996365 0.002958566 0.71696873
## raceindian  0.71369024 0.339566932 1.08781356
```

Compare this model with the no-variable model and all-variable model by LR test and AIC comparison,

```
# LR test
anova(slr_chol0, mlr_chol1, test = "LRT")  # sig. better than no var at all!
```

```
## Analysis of Deviance Table
##
## Model 1: chol ~ 1
## Model 2: chol ~ dbp + race
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       199     278.77
## 2       196     213.40  3   65.373 5.755e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# model with no var at all is called Null Model
anova(mlr_chol, mlr_chol1, test = "LRT")  # no sig. dif with all vars model,
```

```
## Analysis of Deviance Table
## 
## Model 1: chol ~ sbp + dbp + bmi + race
## Model 2: chol ~ dbp + race
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       194     211.34                     
## 2       196     213.40 -2  -2.0593   0.3886
```

```r
# model with 2 vars (dbp & race) is just as good as full model (with all the vars)
# model with all vars is called Saturated Model
```

```r
# AIC
AIC(slr_chol0, mlr_chol1, mlr_chol)
```

```
##           df      AIC
## slr_chol0  2 637.9921
## mlr_chol1  5 590.5459
## mlr_chol   7 592.6065
```

```r
# our final model has the lowest AIC
```

1.3.2.5 Multicollinearity, MC

Multicollinearity is the problem of repetitive/redundant variables – high correlations between predictors. MC is checked by Variance Inflation Factor (VIF). VIF > 10 indicates MC problem.

```r
vif(mlr_chol1)  # all < 10
```

```
##          GVIF Df GVIF^(1/(2*Df))
## dbp  1.132753  1        1.064309
## race 1.132753  2        1.031653
```

1.3.2.6 Interaction, *

Interaction is the predictor variable combination that requires interpretation of regression coefficients separately based on the levels of the predictor (e.g. separate analysis for each race group, Malay vs Chinese vs Indian). This makes interpreting our analysis complicated. So, most of the time, we pray not to have interaction in our regression model.

```r
summary(glm(chol ~ dbp*race, data = coronary))  # dbp*race not sig.
```

```
## 
## Call:
## glm(formula = chol ~ dbp * race, data = coronary)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -2.10485  -0.77524  -0.02423   0.58059   2.74380  
## 
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)  
## (Intercept)        2.11114    0.92803   2.275 0.024008 *
## dbp              0.04650    0.01193   3.897 0.000134 ***
## racechinese      1.95576    1.28477   1.522 0.129572
## raceindian       2.41530    1.25766   1.920 0.056266 .
## dbp:racechinese -0.02033    0.01596  -1.273 0.204376
## dbp:raceindian  -0.02126    0.01529  -1.391 0.165905
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.087348)
##
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 210.95  on 194  degrees of freedom
## AIC: 592.23
##
## Number of Fisher Scoring iterations: 2
```

```
# in R, it is easy to fit interaction by *
# dbp*race will automatically include all vars involved i.e. equal to
# glm(chol ~ dbp + race + dbp:race, data = coronary)
# use : to just include just the interaction
```

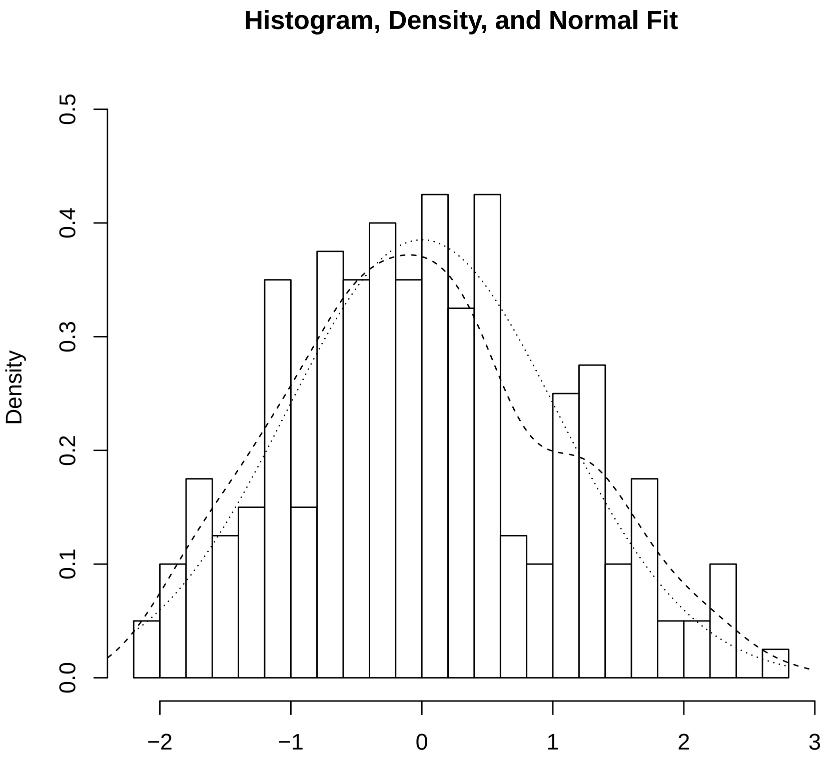
There is no interaction here because the included interaction term was insignificant.

1.3.3 Model fit assessment: Residuals

Histogram

Raw residuals: Normality assumption.

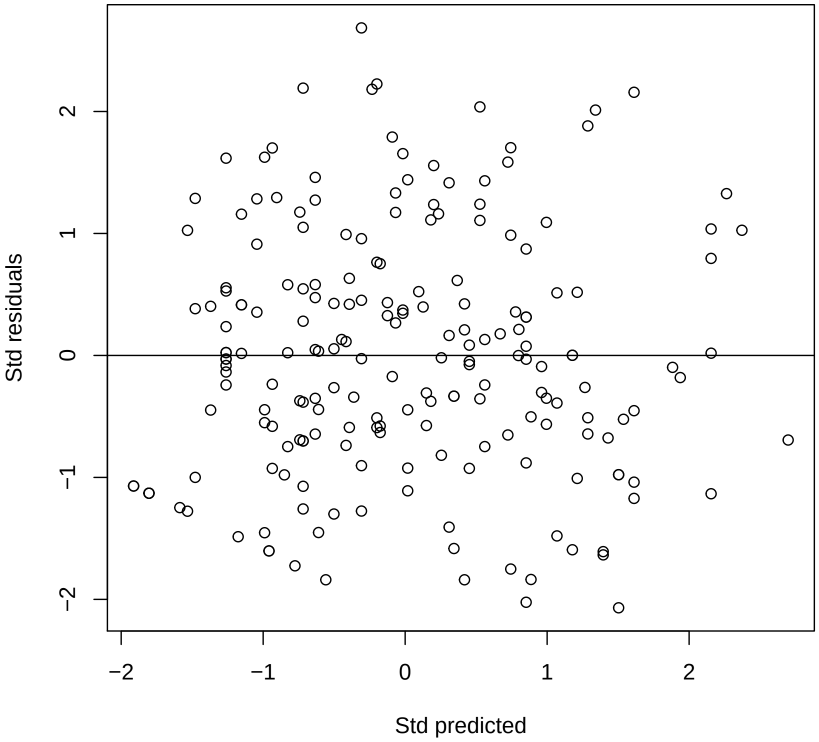
```
rraw_chol = resid(mlr_chol1)  # unstandardized
multi.hist(rraw_chol)
```



Scatter plots

Standardized residuals vs Standardized predicted values: Overall – normality, linearity and equal variance assumptions.

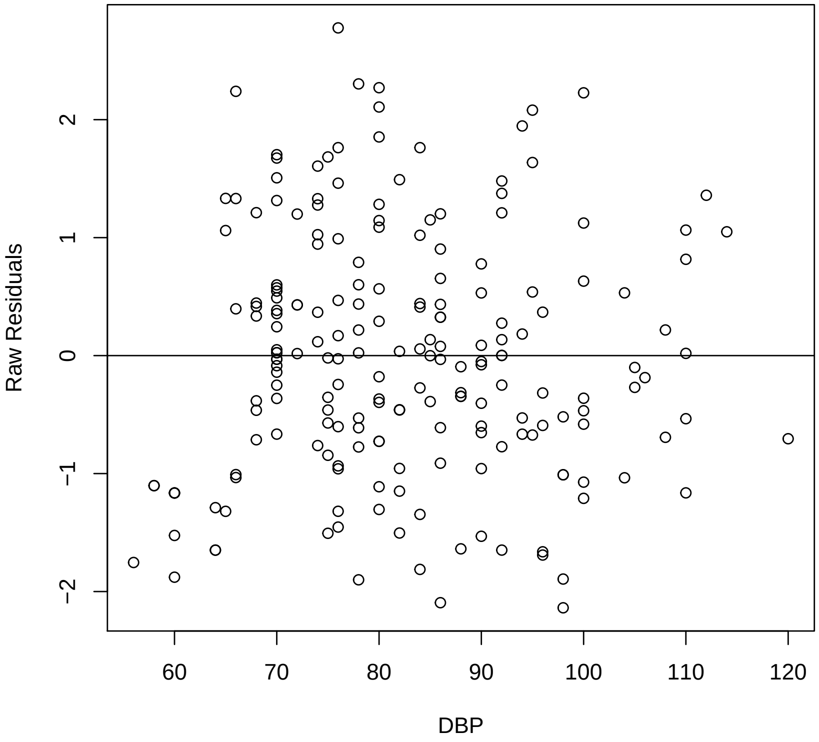
```r
rstd_chol = rstandard(mlr_chol1)  # standardized residuals
pstd_chol = scale(predict(mlr_chol1))  # standardized predicted values
plot(rstd_chol ~ pstd_chol, xlab = "Std predicted", ylab = "Std residuals")
abline(0, 0)  # normal, linear, equal variance
```



The dots should form elliptical/oval shape (normality) and scattered roughly equal above and below the zero line (equal variance). Both these indicate linearity.

Raw residuals vs Numerical predictor by each predictors: Linearity assumption.

```
plot(rraw_chol ~ coronary$dbp, xlab = "DBP", ylab = "Raw Residuals")
abline(0, 0)
```



1.3.4 Interpretation

Now we have decided on our final model, rename the model,

```
# rename the selected model
mlr_chol_final = mlr_chol1
```

and interpret the model,

```
summary(mlr_chol_final)
```

```
##
## Call:
## glm(formula = chol ~ dbp + race, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1378  -0.7068  -0.0289   0.5997   2.7778
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.298028   0.486213   6.783 1.36e-10 ***
## dbp         0.031108   0.006104   5.096 8.14e-07 ***
## racechinese 0.359964   0.182149   1.976 0.049534 *
## raceindian  0.713690   0.190883   3.739 0.000243 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.088777)
##
##     Null deviance: 278.77  on 199  degrees of freedom
## Residual deviance: 213.40  on 196  degrees of freedom
## AIC: 590.55
##
## Number of Fisher Scoring iterations: 2
```

```
Confint(mlr_chol_final)  # 95% CI of the coefficients
```

```
##               Estimate       2.5 %     97.5 %
## (Intercept) 3.29802826 2.345067995 4.25098852
## dbp         0.03110811 0.019143668 0.04307255
## racechinese 0.35996365 0.002958566 0.71696873
## raceindian  0.71369024 0.339566932 1.08781356
```

```
rsq(mlr_chol_final, adj = T)
```

```
## [1] 0.2227869
```

- 1mmHg increase in DBP causes 0.03mmol/L increase in cholestrol, controlling for the effect of race.
- Being Chinese causes 0.36mmol/L increase in cholestrol in comparison to Malay, controlling for the effect of DBP.
- Being Indian causes 0.71mmol/L increase in cholestrol in comparison to Malay, controlling for the effect of DBP.
- DBP and race explains 22.3% variance in cholestrol.

1.3.5 Model equation

Cholestrol level in mmol/L can be predicted by its predictors as given by,

$$chol = 3.30 + 0.03 \times dbp + 0.36 \times race\ (chinese) + 0.71 \times race\ (indian)$$

1.3.6 Prediction

It is easy to predict in R using our fitted model above. First we view the predicted values for our sample,

```
coronary$pred_chol = predict(mlr_chol_final)
head(coronary)
```

```
##   sbp dbp   chol  bmi   race gender pred_chol
## 1 106  68 6.5725 38.9 indian  woman  6.127070
## 2 130  78 6.3250 37.8  malay  woman  5.724461
## 3 136  84 5.9675 40.5  malay  woman  5.911109
## 4 138 100 7.0400 37.6  malay  woman  6.408839
## 5 115  85 6.6550 40.3 indian    man  6.655908
## 6 124  72 5.9675 37.6  malay    man  5.537812
```

Now let us try predicting for any values for `dbp` and `race`,

```
str(coronary[c("dbp", "race")])
```

```
## 'data.frame':    200 obs. of  2 variables:
##  $ dbp : num  68 78 84 100 85 72 80 70 85 70 ...
##  $ race: Factor w/ 3 levels "malay","chinese",..: 3 1 1 1 3 1 1 2 2 2 ...
```

```
# simple, dbp = 90, race = indian
predict(mlr_chol_final, list(dbp = 90, race = "indian"))
```

```
##        1
## 6.811448
```

More data points

```
new_data = data.frame(dbp = c(90, 90, 90), race = c("malay", "chinese", "indian"))
new_data
```

```
##   dbp    race
## 1  90   malay
## 2  90 chinese
## 3  90  indian
```

```
predict(mlr_chol_final, new_data)
```

```
##        1        2        3
## 6.097758 6.457722 6.811448
```

```
new_data$pred_chol = predict(mlr_chol_final, new_data)
new_data
```

```
##   dbp    race pred_chol
## 1  90   malay  6.097758
## 2  90 chinese  6.457722
## 3  90  indian  6.811448
```

1.4 Exercises

1. Present the results in a table (follow Arifin et al. (2016))
2. Obtain the coefficient for 5mmHg increase in DBP.
3. Add `age` to the multivariable model. What happens?

Chapter 2

Logistic Regression

2.1 Introduction

1. Statistical method to model relationship between:
   - outcome: binary categorical variable.
   - predictors/independent variables: numerical, categorical variables.
2. A type of Generalized Linear Models (GLMs).
3. Basically, the relationship is structured as follows,

$$binary\ outcome = numerical\ predictors + categorical\ predictors$$

more accurately, the *logistic* relationship structure,

$$log_e\left(\frac{proportion}{1 - proportion}\right) = numerical\ predictors + categorical\ predictors$$

We turned the binary outcome into proportion (p) of having the outcome. log_e is the *natural log*, sometimes written as *ln*.

The part, $\frac{p}{1-p}$ is known as *odds*.

2.2 Odds ratio vs relative risk

Association analysis for cross-tabulation of a binary factor and its outcome can be expressed as odds ratio.

- Odds is a measure of chance of disease occurence in a specified group,

$$Odds = \frac{n_{disease}}{n_{no\ disease}}$$

- Odds ratio, OR is the ratio between the odds of two groups; the group with the risk factor and the group without the risk factor,

$$Odds\ ratio, OR = \frac{Odds_{factor}}{Odds_{no\ factor}}$$

Odds ratio can be calculated for cohort, cross-sectional and case-control studied because it does not imply a cause-effect association, but only plain association.

In epidemiology, it is common to describe the association between a risk factor and a disease in term of risk and relative risk.

- Risk is a measure of chance of disease occurence in a specified group, calculated as

$$Risk = \frac{n_{disease}}{n_{group}}$$

- Relative risk is the ration between the risk in the group with the factor and the risk in the group without the risk factor,

$$Relative\ risk, RR = \frac{Risk_{factor}}{Risk_{no\ factor}}$$

  It is only approriate to calculate risk and relative risk for cohort studies, because the cause-effect relationship is well defined.

OR is a good approximation of RR whenever the disease is rare. Rare diseases are commonly studied using case-control studies, thus the use of ORs are justified.

As an example, we can calculate odds, OR, risk and RR from the following table.

Table 2.1: Smoker vs lung cancer

	Lung cancer	No lung cancer	Marginal total	Odds	Risk
Smoker	20	12	32	20/12 = 1.667	20/32 = 0.625
Non smoker	95	73	168	95/73 = 1.301	95/168 = 0.565

Thus OR and RR equal,

$$OR = 1.667/1.301 = 1.281$$

$$RR = 0.625/0.565 = 1.106$$

2.3 Simple logistic regression (SLogR)

About SLogR

1. Model relatioship between:
   - outcome: binary categorical variable.
   - a predictor: numerical or binary categorical variable.
2. Formula,

$$log_e\left(\frac{p}{1-p}\right) = intercept + coefficient \times numerical/binary\ predictor$$

   or in a proper equation form,

$$log_e\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1$$

3. Odds ratio is easily obtained from a logistic regression,

$$OR_1 = e^{\beta_1}$$

4. p – proportion/probability. To obtain p,

$$p = \frac{e^{\beta_0+\beta_1 x_1}}{1+e^{\beta_0+\beta_1 x_1}}$$

But as we will see later, this can be easily obtained in R.

Analysis

```
# library
library(foreign)
library(epiDisplay)
library(psych)
library(lattice)
library(rsq)
library(MASS)
library(car)
```

```
# data
coronary = read.dta("coronary.dta")
str(coronary)
```

```
## 'data.frame':    200 obs. of  9 variables:
##  $ id    : num  1 14 56 61 62 64 69 108 112 134 ...
##  $ cad   : Factor w/ 2 levels "no cad","cad": 1 1 1 1 1 1 2 1 1 1 ...
##  $ sbp   : num  106 130 136 138 115 124 110 112 138 104 ...
##  $ dbp   : num  68 78 84 100 85 72 80 70 85 70 ...
##  $ chol  : num  6.57 6.33 5.97 7.04 6.66 ...
##  $ age   : num  60 34 36 45 53 43 44 50 43 48 ...
##  $ bmi   : num  38.9 37.8 40.5 37.6 40.3 ...
##  $ race  : Factor w/ 3 levels "malay","chinese",..: 3 1 1 1 3 1 1 2 2 2 ...
##  $ gender: Factor w/ 2 levels "woman","man": 1 1 1 1 2 2 2 1 1 2 ...
##  - attr(*, "datalabel")= chr "Written by R.               "
##  - attr(*, "time.stamp")= chr ""
##  - attr(*, "formats")= chr  "%9.0g" "%9.0g" "%9.0g" "%9.0g" ...
##  - attr(*, "types")= int  100 108 100 100 100 100 100 108 108
##  - attr(*, "val.labels")= chr  "" "cad" "" "" ...
##  - attr(*, "var.labels")= chr  "id" "cad" "sbp" "dbp" ...
##  - attr(*, "version")= int 7
##  - attr(*, "label.table")=List of 3
##   ..$ cad   : Named int  1 2
##   .. ..- attr(*, "names")= chr  "no cad" "cad"
##   ..$ race  : Named int  1 2 3
##   .. ..- attr(*, "names")= chr  "malay" "chinese" "indian"
##   ..$ gender: Named int  1 2
##   .. ..- attr(*, "names")= chr  "woman" "man"
```

2.3.1 Data exploration

2.3.1.1 Descriptive statistics

```
codebook(coronary[c("cad", "gender")])
```

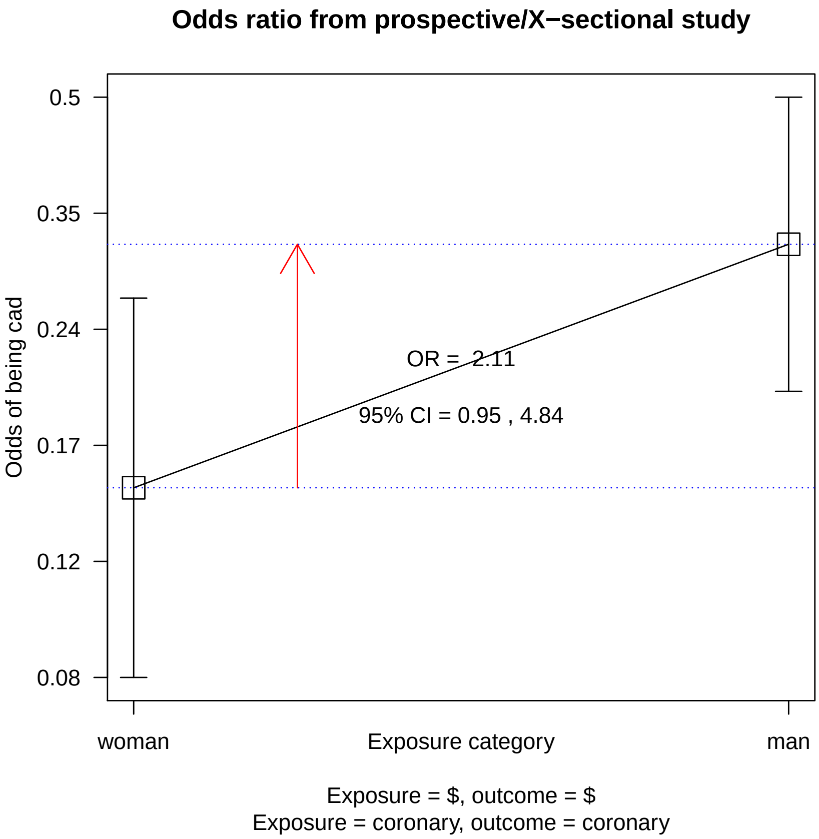
```
##
```

```
##
##
## cad    :
##         Frequency Percent
## no cad        163    81.5
## cad            37    18.5
##
##  ==================
## gender    :
##        Frequency Percent
## woman        100      50
## man          100      50
##
##  ==================
```

```
table(coronary$gender, coronary$cad)
```

```
##
##          no cad cad
##   woman      87  13
##   man        76  24
```

```
cc(coronary$cad, coronary$gender)  # plain OR
```



```
##
##             coronary$gender
## coronary$cad woman man Total
##       no cad    87  76   163
##       cad       13  24    37
##       Total    100 100   200
##
## OR =  2.11
## 95% CI =  1.01, 4.44
## Chi-squared = 4.01, 1 d.f., P value = 0.045
## Fisher's exact test (2-sided) P value = 0.068
```

2.3.2 Univariable

Fit model,

```
# model: cad ~ gender
slg_cad = glm(cad ~ gender, data = coronary, family = binomial)
summary(slg_cad)
```

```
##
## Call:
## glm(formula = cad ~ gender, family = binomial, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7409  -0.7409  -0.5278  -0.5278   2.0200
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.9010     0.2973  -6.393 1.63e-10 ***
## genderman     0.7483     0.3785   1.977    0.048 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 187.49  on 198  degrees of freedom
## AIC: 191.49
##
## Number of Fisher Scoring iterations: 4
```

```
Confint(slg_cad)  # coeff.
```

```
##               Estimate       2.5 %    97.5 %
## (Intercept) -1.9009588 -2.53093234 -1.355540
## genderman    0.7482793  0.02044525  1.514515
```

```
exp(Confint(slg_cad))  # OR
```

```
##              Estimate      2.5 %    97.5 %
## (Intercept) 0.1494253 0.07958479 0.2578081
## genderman   2.1133603 1.02065568 4.5472149
```

Focus on:

- Coefficient, β and OR.
- 95% CI.

- *P*-value.

2.3.3 Interpretation

We are most interested in the OR,

- Man is at 2.11 odds of having coronary artery disease (CAD) as compared to woman.

Be careful with the terms; odds vs risk!

2.3.4 Model equation

$$log_e\left(\frac{p_{cad}}{1-p_{cad}}\right) = -1.90 + 0.75 \times gender\ (man)$$

$$p_{cad} = \frac{e^{-1.9+0.75\times gender\ (man)}}{1+e^{-1.90+0.75\times gender\ (man)}}$$

Note: Don't scratch your head.

2.4 Multiple logistic regression (MLogR)

1. Model relatioship between:
    - outcome: binary categorical variable.
    - predictors: numerical, categorical variables.
2. Formula,

$$\begin{aligned} log_e\left(\frac{p}{1-p}\right) &= intercept + coefficients \times numerical\ predictors \\ &+ coefficients \times categorical\ predictors \end{aligned}$$

or in a nicer form,

$$log_e\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k$$

where we have k predictors.

Whenever the predictor is a categorical variable with more than two levels, remember to consider dummy (binary) variable(s).

Analysis

```
str(coronary)
```

```
## 'data.frame':    200 obs. of  9 variables:
##  $ id    : num  1 14 56 61 62 64 69 108 112 134 ...
##  $ cad   : Factor w/ 2 levels "no cad","cad": 1 1 1 1 1 1 2 1 1 1 ...
##  $ sbp   : num  106 130 136 138 115 124 110 112 138 104 ...
##  $ dbp   : num  68 78 84 100 85 72 80 70 85 70 ...
##  $ chol  : num  6.57 6.33 5.97 7.04 6.66 ...
##  $ age   : num  60 34 36 45 53 43 44 50 43 48 ...
##  $ bmi   : num  38.9 37.8 40.5 37.6 40.3 ...
##  $ race  : Factor w/ 3 levels "malay","chinese",..: 3 1 1 1 3 1 1 2 2 2 ...
##  $ gender: Factor w/ 2 levels "woman","man": 1 1 1 1 2 2 2 1 1 2 ...
##  - attr(*, "datalabel")= chr "Written by R.               "
##  - attr(*, "time.stamp")= chr ""
##  - attr(*, "formats")= chr  "%9.0g" "%9.0g" "%9.0g" "%9.0g" ...
##  - attr(*, "types")= int  100 108 100 100 100 100 100 108 108
##  - attr(*, "val.labels")= chr  "" "cad" "" "" ...
##  - attr(*, "var.labels")= chr  "id" "cad" "sbp" "dbp" ...
##  - attr(*, "version")= int 7
##  - attr(*, "label.table")=List of 3
##   ..$ cad   : Named int  1 2
##   .. ..- attr(*, "names")= chr  "no cad" "cad"
##   ..$ race  : Named int  1 2 3
##   .. ..- attr(*, "names")= chr  "malay" "chinese" "indian"
##   ..$ gender: Named int  1 2
##   .. ..- attr(*, "names")= chr  "woman" "man"
```

```r
coronary = subset(coronary, select = -id) # remove id
```

2.4.1 Data exploration

2.4.1.1 Descriptive statistics

By CAD status,

```r
by(subset(coronary, select = c(sbp, dbp, chol, age, bmi)), coronary$cad, summ)
```

```
## coronary$cad: no cad
## 
## No. of observations = 163
## 
##   Var. name obs. mean   median  s.d.   min.   max.
## 1 sbp       163  127.84 124     19.14  88     187
## 2 dbp       163  80.8   80      12.61  56     120
## 3 chol      163  6.1    6.05    1.17   4      9.35
## 4 age       163  46.79  47      7.4    32     62
## 5 bmi       163  37.58  38      2.48   28.99  41.2
## --------------------------------------------------------
## coronary$cad: cad
## 
## No. of observations = 37
## 
##   Var. name obs. mean   median  s.d.   min.   max.
## 1 sbp       37   140.49 138     19.67  100    178
## 2 dbp       37   88.97  90      12.17  70     114
## 3 chol      37   6.65   6.66    1.17   4.12   9.05
## 4 age       37   49.7   50      6.66   35     61
## 5 bmi       37   36.86  37.14   3.39   31     45.03
```

```r
by(subset(coronary, select = c(race, gender)), coronary$cad, codebook)
```

```
## 
## 
## 
## race      :
```

```
##         Frequency Percent
## malay          60    36.8
## chinese        52    31.9
## indian         51    31.3
## 
##  ==================
## gender    :
##       Frequency Percent
## woman        87    53.4
## man          76    46.6
## 
##  ==================
## 
## 
## 
## race      :
##         Frequency Percent
## malay          13    35.1
## chinese        12    32.4
## indian         12    32.4
## 
##  ==================
## gender    :
##       Frequency Percent
## woman        13    35.1
## man          24    64.9
## 
##  ==================
```

```
## coronary$cad: no cad
## NULL
## --------------------------------------------------------
## coronary$cad: cad
## NULL
```

2.4.2 Univariable

Perform SLogR for `sbp, dbp, chol, age, bmi, race` and `gender` on your own. Now, we want to determine which variables are worthwhile to include in the multivariable models.

We want to screen variables with P-values < 0.25 to be included in MLogR. Obtaining the P-values for each variable is easy by LR test,

```
slg_cad0 = glm(cad ~ 1, data = coronary, family = binomial)
summary(slg_cad0)
```

```
## 
## Call:
## glm(formula = cad ~ 1, family = binomial, data = coronary)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -0.6396  -0.6396  -0.6396  -0.6396   1.8371  
## 
## Coefficients:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.4828     0.1821  -8.143 3.86e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 191.56  on 199  degrees of freedom
## AIC: 193.56
##
## Number of Fisher Scoring iterations: 4
```

```
names(coronary)
```

```
## [1] "cad"    "sbp"    "dbp"    "chol"   "age"    "bmi"    "race"   "gender"
```

```
add1(slg_cad0, scope = ~ sbp + dbp + chol + age + bmi + race + gender,
     test = "LRT")
```

```
## Single term additions
##
## Model:
## cad ~ 1
##        Df Deviance    AIC     LRT  Pr(>Chi)
## <none>      191.56 193.56
## sbp     1   179.62 183.62 11.9339 0.0005512 ***
## dbp     1   179.62 183.62 11.9333 0.0005514 ***
## chol    1   185.04 189.04  6.5187 0.0106747 *
## age     1   186.72 190.72  4.8346 0.0278945 *
## bmi     1   189.38 193.38  2.1811 0.1397120
## race    2   191.52 197.52  0.0385 0.9809448
## gender  1   187.49 191.49  4.0631 0.0438292 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All variables are $< .25$ except `race`. We will include all variables in MLogR except `race`.

2.4.2.1 Multivariable

Perform MLogR with ALL selected variables,

```
# all
mlg_cad = glm(cad ~ sbp + dbp + chol + age + bmi + gender,
              data = coronary, family = binomial)
summary(mlg_cad)
```

```
##
## Call:
## glm(formula = cad ~ sbp + dbp + chol + age + bmi + gender, family = binomial,
##     data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3919  -0.6212  -0.4947  -0.3659   2.2476
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.350564   3.217917  -1.663   0.0964 .
## sbp          0.010748   0.017583   0.611   0.5410
## dbp          0.026556   0.026789   0.991   0.3215
## chol         0.136521   0.186445   0.732   0.4640
## age          0.009897   0.032090   0.308   0.7578
## bmi         -0.041313   0.068023  -0.607   0.5436
## genderman    0.683946   0.403712   1.694   0.0902 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 173.33  on 193  degrees of freedom
## AIC: 187.33
##
## Number of Fisher Scoring iterations: 4
```

At this point, focus on:

- Coefficients, βs.
- P-values.

For model fit,

- Akaike Information Criterion, AIC – for comparison with other models. This is not useful alone, but for comparison with other models. The model with the lowest AIC is the best model.

2.4.2.2 Stepwise

As you can see, not all variables are significant. How to select? We proceed with stepwise automatic selection,

```
# both
mlg_cad_stepboth = step(mlg_cad, direction = "both")
```

```
## Start:  AIC=187.33
## cad ~ sbp + dbp + chol + age + bmi + gender
##
##          Df Deviance    AIC
## - age     1   173.43 185.43
## - bmi     1   173.70 185.70
## - sbp     1   173.70 185.70
## - chol    1   173.87 185.87
## - dbp     1   174.33 186.33
## <none>        173.33 187.33
## - gender  1   176.28 188.28
##
## Step:  AIC=185.43
## cad ~ sbp + dbp + chol + bmi + gender
##
##          Df Deviance    AIC
## - bmi     1   173.78 183.78
## - sbp     1   173.95 183.95
## - chol    1   174.09 184.09
## - dbp     1   174.40 184.40
## <none>        173.43 185.43
## - gender  1   176.61 186.61
## + age     1   173.33 187.33
##
## Step:  AIC=183.78
## cad ~ sbp + dbp + chol + gender
##
##          Df Deviance    AIC
## - sbp     1   174.26 182.26
## - chol    1   174.53 182.53
## - dbp     1   174.91 182.91
## <none>        173.78 183.78
## - gender  1   177.09 185.09
## + bmi     1   173.43 185.43
## + age     1   173.70 185.70
##
## Step:  AIC=182.26
## cad ~ dbp + chol + gender
##
##          Df Deviance    AIC
## - chol    1   175.21 181.21
## <none>        174.26 182.26
## + sbp     1   173.78 183.78
## - gender  1   177.86 183.86
## + bmi     1   173.95 183.95
## + age     1   174.05 184.05
## - dbp     1   181.87 187.87
##
## Step:  AIC=181.2
## cad ~ dbp + gender
##
##          Df Deviance    AIC
## <none>        175.21 181.21
## + chol    1   174.26 182.26
## + sbp     1   174.53 182.53
## + age     1   174.74 182.74
## + bmi     1   174.80 182.80
## - gender  1   179.62 183.62
## - dbp     1   187.49 191.49
```

```
summary(mlg_cad_stepboth)  # cad ~ dbp + gender
```

```
##
## Call:
## glm(formula = cad ~ dbp + gender, family = binomial, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4520  -0.6508  -0.5249  -0.3643   2.3337
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.12046    1.31667  -4.648 3.34e-06 ***
## dbp          0.04950    0.01463   3.383 0.000717 ***
## genderman    0.80573    0.39084   2.062 0.039253 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 175.20  on 197  degrees of freedom
## AIC: 181.2
##
## Number of Fisher Scoring iterations: 4
```

```
# forward
mlg_cad_stepforward = step(slg_cad0,
                           scope = ~ sbp + dbp + chol + age + bmi + gender,
                           direction = "forward")
```

```
## Start:  AIC=193.56
## cad ~ 1
##
##          Df Deviance    AIC
## + sbp     1   179.62 183.62
## + dbp     1   179.62 183.62
## + chol    1   185.04 189.04
## + age     1   186.72 190.72
## + gender  1   187.49 191.49
## + bmi     1   189.38 193.38
## <none>        191.56 193.56
##
## Step:  AIC=183.62
## cad ~ sbp
##
##          Df Deviance    AIC
## + gender  1   176.00 182.00
## <none>        179.62 183.62
## + chol    1   177.86 183.86
## + dbp     1   178.52 184.52
## + bmi     1   178.80 184.80
## + age     1   179.09 185.09
##
## Step:  AIC=182
## cad ~ sbp + gender
##
##        Df Deviance    AIC
## <none>      176.00 182.00
## + dbp   1   174.53 182.53
## + chol  1   174.91 182.91
## + bmi   1   175.32 183.32
## + age   1   175.84 183.84
```

```
summary(mlg_cad_stepforward)  # cad ~ sbp + gender
```

```
##
## Call:
## glm(formula = cad ~ sbp + gender, family = binomial, data = coronary)
##
```

```
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.3815  -0.6348  -0.5069  -0.3871   2.4379  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -5.973612   1.295822  -4.610 4.03e-06 ***
## sbp          0.030546   0.009222   3.312 0.000925 ***
## genderman    0.729389   0.389404   1.873 0.061056 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 176.00  on 197  degrees of freedom
## AIC: 182
## 
## Number of Fisher Scoring iterations: 4
```

```r
# backward
mlg_cad_stepback = step(mlg_cad, direction = "backward")
```

```
## Start:  AIC=187.33
## cad ~ sbp + dbp + chol + age + bmi + gender
## 
##          Df Deviance    AIC
## - age     1   173.43 185.43
## - bmi     1   173.70 185.70
## - sbp     1   173.70 185.70
## - chol    1   173.87 185.87
## - dbp     1   174.33 186.33
## <none>        173.33 187.33
## - gender  1   176.28 188.28
## 
## Step:  AIC=185.43
## cad ~ sbp + dbp + chol + bmi + gender
## 
##          Df Deviance    AIC
## - bmi     1   173.78 183.78
## - sbp     1   173.95 183.95
## - chol    1   174.09 184.09
## - dbp     1   174.40 184.40
## <none>        173.43 185.43
## - gender  1   176.61 186.61
## 
## Step:  AIC=183.78
## cad ~ sbp + dbp + chol + gender
## 
##          Df Deviance    AIC
## - sbp     1   174.26 182.26
## - chol    1   174.53 182.53
## - dbp     1   174.91 182.91
## <none>        173.78 183.78
## - gender  1   177.09 185.09
```

```
##
## Step:  AIC=182.26
## cad ~ dbp + chol + gender
##
##          Df Deviance    AIC
## - chol    1   175.21 181.21
## <none>        174.26 182.26
## - gender  1   177.86 183.86
## - dbp     1   181.87 187.87
##
## Step:  AIC=181.2
## cad ~ dbp + gender
##
##          Df Deviance    AIC
## <none>        175.21 181.21
## - gender  1   179.62 183.62
## - dbp     1   187.49 191.49
```

```
summary(mlg_cad_stepback)  # cad ~ dbp + gender
```

```
##
## Call:
## glm(formula = cad ~ dbp + gender, family = binomial, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4520  -0.6508  -0.5249  -0.3643   2.3337
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.12046    1.31667  -4.648 3.34e-06 ***
## dbp          0.04950    0.01463   3.383 0.000717 ***
## genderman    0.80573    0.39084   2.062 0.039253 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 175.20  on 197  degrees of freedom
## AIC: 181.2
##
## Number of Fisher Scoring iterations: 4
```

Looking at all these results, there are two competing models:

> `cad ~ dbp + gender` (mlg_cad_stepboth and mlg_cad_stepback) vs `cad ~ sbp + gender` (mlg_cad_stepforward)

We compare the AICs,

```
AIC(mlg_cad_stepboth, mlg_cad_stepforward)
```

```
##                     df      AIC
## mlg_cad_stepboth     3 181.2047
## mlg_cad_stepforward  3 181.9997
```

```r
# mlg_cad_stepboth: cad ~ dbp + gender, gives the lowest AIC
# mlg_cad_stepforward: cad ~ sbp + gender, gives insig. p-value to gender
```

`cad ~ dbp + gender` has the lowest AIC, which we now name as `mlg_cad1`,

```r
# mlg_cad1: cad ~ dbp + gender
mlg_cad1 = glm(cad ~ dbp + gender, data = coronary, family = binomial)
summary(mlg_cad1)
```

```
##
## Call:
## glm(formula = cad ~ dbp + gender, family = binomial, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4520  -0.6508  -0.5249  -0.3643   2.3337
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.12046    1.31667  -4.648 3.34e-06 ***
## dbp          0.04950    0.01463   3.383 0.000717 ***
## genderman    0.80573    0.39084   2.062 0.039253 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 175.20  on 197  degrees of freedom
## AIC: 181.2
##
## Number of Fisher Scoring iterations: 4
```

2.4.2.3 Confounder

If we include a variable and it causes notable change ($> 20\%$) in the coefficients of other variables, it is a confounder. When the confounder is significant and the main effect variable is also significant, we keep the confounder in the model.

Formula for % change,

```
100 * (model_small - model_large) / model_large
```

Hosmer et al. (2013)

Now we want add back all possible variables and variables removed before.

```r
# + age, common demographic confounder
summary(update(mlg_cad1, . ~ . + age))  # longer codes
```

```
##
## Call:
## glm(formula = cad ~ dbp + gender + age, family = binomial, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4645  -0.6346  -0.5058  -0.3674   2.3714
```

```
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -6.70568    1.58824  -4.222 2.42e-05 ***
## dbp          0.04528    0.01578   2.869  0.00412 ** 
## genderman    0.75629    0.39653   1.907  0.05649 .  
## age          0.02017    0.02945   0.685  0.49345    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 174.74  on 196  degrees of freedom
## AIC: 182.74
##
## Number of Fisher Scoring iterations: 4
```

```
coef(update(mlg_cad1, . ~ . + age))  # no need to save into objects
```

```
## (Intercept)         dbp   genderman         age 
## -6.70568442  0.04527739  0.75628533  0.02016735
```

```
coef(mlg_cad1)
```

```
## (Intercept)         dbp   genderman 
## -6.12046337  0.04950439  0.80572747
```

```
100 * (coef(mlg_cad1) - coef(update(mlg_cad1, . ~ . + age))[1:3]) /
  coef(update(mlg_cad1, . ~ . + age))[1:3]
```

```
## (Intercept)         dbp   genderman 
##   -8.727238    9.335785    6.537497
```

```
# < 20% change
```

```
# + chol
summary(update(mlg_cad1, . ~ . + chol))
```

```
##
## Call:
## glm(formula = cad ~ dbp + gender + chol, family = binomial, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max  
## -1.3923  -0.6290  -0.5147  -0.3633   2.3033  
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -6.65749    1.45211  -4.585 4.55e-06 ***
## dbp          0.04314    0.01598   2.700  0.00693 ** 
## genderman    0.74112    0.39642   1.870  0.06155 .  
## chol         0.17498    0.17966   0.974  0.33009    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 174.26  on 196  degrees of freedom
## AIC: 182.26
##
## Number of Fisher Scoring iterations: 4
```

```
coef(update(mlg_cad1, . ~ . + chol))
```

```
## (Intercept)         dbp   genderman        chol
## -6.65749252  0.04313952  0.74112395  0.17498152
```

```
coef(mlg_cad1)
```

```
## (Intercept)         dbp   genderman
## -6.12046337  0.04950439  0.80572747
```

```
100 * (coef(mlg_cad1) - coef(update(mlg_cad1, . ~ . + chol))[1:3]) /
  coef(update(mlg_cad1, . ~ . + chol))[1:3]  # [1:3] select vars, exclude new var
```

```
## (Intercept)         dbp   genderman
##   -8.066538   14.754162    8.716965
```

```
# < 20% change
```

```
# + bmi
summary(update(mlg_cad1, . ~ . + bmi))
```

```
##
## Call:
## glm(formula = cad ~ dbp + gender + bmi, family = binomial, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4030  -0.6506  -0.5133  -0.3479   2.3236
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.35227    3.06324  -1.421  0.15537
## dbp          0.04766    0.01489   3.200  0.00137 **
## genderman    0.78721    0.39220   2.007  0.04473 *
## bmi         -0.04300    0.06760  -0.636  0.52471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 174.80  on 196  degrees of freedom
## AIC: 182.8
##
## Number of Fisher Scoring iterations: 4
```

```
coef(update(mlg_cad1, . ~ . + bmi))
```

```
## (Intercept)         dbp   genderman         bmi
## -4.35226609  0.04766414  0.78721006 -0.04300184
```

```
coef(mlg_cad1)
```

```
## (Intercept)         dbp   genderman
## -6.12046337  0.04950439  0.80572747
```

```
100 * (coef(mlg_cad1) - coef(update(mlg_cad1, . ~ . + bmi))[1:3]) /
  coef(update(mlg_cad1, . ~ . + bmi))[1:3]
```

```
## (Intercept)         dbp   genderman
##   40.627049    3.860871    2.352282
```

```
# < 20% change. Again ignore the intercept.
```

```
# + race
summary(update(mlg_cad1, . ~ . + race))
```

```
##
## Call:
## glm(formula = cad ~ dbp + gender + race, family = binomial, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4413  -0.6424  -0.5080  -0.3140   2.5925
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.72321    1.41307  -4.758 1.96e-06 ***
## dbp          0.06014    0.01653   3.637 0.000276 ***
## genderman    0.92006    0.40356   2.280 0.022615 *
## racechinese -0.35168    0.47619  -0.739 0.460188
## raceindian  -0.81170    0.53230  -1.525 0.127284
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 172.78  on 195  degrees of freedom
## AIC: 182.78
##
## Number of Fisher Scoring iterations: 5
```

```
coef(update(mlg_cad1, . ~ . + race))
```

```
## (Intercept)         dbp   genderman racechinese  raceindian
## -6.72320622  0.06013888  0.92006448 -0.35168228 -0.81170429
```

```
coef(mlg_cad1)
```

```
## (Intercept)         dbp   genderman
## -6.12046337  0.04950439  0.80572747
```

```
100 * (coef(mlg_cad1) - coef(update(mlg_cad1, . ~ . + race))[1:3]) /
  coef(update(mlg_cad1, . ~ . + race))[1:3]
```

```
## (Intercept)         dbp   genderman
##    -8.96511   -17.68322   -12.42707
```

```
# < 20% change
```

Lastly we add sbp, which is known to relate to dbp,

```
# + sbp
summary(update(mlg_cad1, . ~ . + sbp))
```

```
##
## Call:
## glm(formula = cad ~ dbp + gender + sbp, family = binomial, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4895  -0.6367  -0.5089  -0.3598   2.3310
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.41803    1.36911  -4.688 2.76e-06 ***
## dbp          0.03136    0.02618   1.198   0.2309
## genderman    0.77165    0.39309   1.963   0.0496 *
## sbp          0.01386    0.01672   0.829   0.4070
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 174.53  on 196  degrees of freedom
## AIC: 182.53
##
## Number of Fisher Scoring iterations: 4
```

```
coef(update(mlg_cad1, . ~ . + sbp))
```

```
## (Intercept)         dbp   genderman         sbp
## -6.41803436  0.03136062  0.77165487  0.01386275
```

```
coef(mlg_cad1)
```

```
## (Intercept)         dbp   genderman
## -6.12046337  0.04950439  0.80572747
```

```
100 * (coef(mlg_cad1) - coef(update(mlg_cad1, . ~ . + sbp))[1:3]) / coef(update(mlg_cad1, . ~ . + sbp))
```

```
## (Intercept)         dbp   genderman
##   -4.636482   57.855257    4.415523
```

```
# > 20% change
```

There is $> 20\%$ change in dbp coefficient, thus sbp is a possible confounder! However, inclusion of sbp causes insignificant P-values for both dbp and sbp. Thus we investigate further the relationship between dbp and sbp by simple correlation,

```
cor(coronary$sbp, coronary$dbp)
```

```
## [1] 0.8277225
```

Both are highly correlated, this actually may fall under multicollinearity (MC) issue below. This is not a plain confounding issue. MC issue will be explained further below. In MC issue, the solution will be that we may choose to include either of the variables, not both. But in our case, in the model with `sbp + gender`, the gender was insignificant, thus we prefer `dbp + gender` model.

Our chosen model:

```
mlg_cad1: cad ~ dbp + gender
```

```
summary(mlg_cad1)
```

```
##
## Call:
## glm(formula = cad ~ dbp + gender, family = binomial, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4520  -0.6508  -0.5249  -0.3643   2.3337
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.12046    1.31667  -4.648 3.34e-06 ***
## dbp          0.04950    0.01463   3.383 0.000717 ***
## genderman    0.80573    0.39084   2.062 0.039253 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 175.20  on 197  degrees of freedom
## AIC: 181.2
##
## Number of Fisher Scoring iterations: 4
```

```
Confint(mlg_cad1)  # 95% CI of the coefficients
```

```
##                Estimate       2.5 %      97.5 %
## (Intercept) -6.12046337 -8.83143505 -3.63733576
## dbp          0.04950439  0.02153556  0.07927883
## genderman    0.80572747  0.05380813  1.59635398
```

Compare this model with the no-variable model and all-variable model by LR test and AIC comparison,

```
# LR test
anova(slg_cad0, mlg_cad1, test = "LRT")  # sig. better than no var at all,
```

```
## Analysis of Deviance Table
##
## Model 1: cad ~ 1
## Model 2: cad ~ dbp + gender
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       199     191.56
## 2       197     175.21  2   16.352 0.0002814 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# i.e. the Null Model
anova(mlg_cad, mlg_cad1, test = "LRT")  # no sig. dif with all vars model,
## Analysis of Deviance Table
## 
## Model 1: cad ~ sbp + dbp + chol + age + bmi + gender
## Model 2: cad ~ dbp + gender
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       193     173.33                     
## 2       197     175.21 -4   -1.872   0.7593
```

```
# model with 2 vars (dbp & gender) is just as good as full model (with all the vars),
# i.e. the Saturated Model
```

```
# AIC
AIC(slg_cad0, mlg_cad1, mlg_cad)
```

```
##          df      AIC
## slg_cad0  1 193.5565
## mlg_cad1  3 181.2047
## mlg_cad   7 187.3327
```

```
# our final model has the lowest AIC
```

2.4.2.4 Multicollinearity, MC

Multicollinearity is the problem of redundant variables, in other words, high correlations between predictors. For logistic regression, this is checked by looking at the estimates and standard errors, SEs. Whenever SE is larger than the estimate, this may point to an MC problem. But how large is large? Relatively large, this is not mentioned specifically in Hosmer et al. (2013). My own guess is that the ratio between SE:estimate should be < 1.

Sometimes, the estimates are unusually large, i.e. indicates very large ORs. This is illogical – also indicates an MC problem.

Again we look at our mlg_cad1 model,

```
# mlg_cad1: cad ~ dbp + gender
summary(mlg_cad1)
```

```
## 
## Call:
## glm(formula = cad ~ dbp + gender, family = binomial, data = coronary)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.4520  -0.6508  -0.5249  -0.3643   2.3337  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -6.12046    1.31667  -4.648 3.34e-06 ***
## dbp          0.04950    0.01463   3.383 0.000717 ***
## genderman    0.80573    0.39084   2.062 0.039253 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 175.20  on 197  degrees of freedom
## AIC: 181.2
##
## Number of Fisher Scoring iterations: 4
```

Fortunately, all SEs < estimates/coefficients.

Now we have a relook at the `sbp` problem above,

```r
# mlg_cad1 + sbp : cad ~ dbp + gender + sbp
summary(update(mlg_cad1, . ~ . + sbp))
```

```
##
## Call:
## glm(formula = cad ~ dbp + gender + sbp, family = binomial, data = coronary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4895  -0.6367  -0.5089  -0.3598   2.3310
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.41803    1.36911  -4.688 2.76e-06 ***
## dbp          0.03136    0.02618   1.198   0.2309
## genderman    0.77165    0.39309   1.963   0.0496 *
## sbp          0.01386    0.01672   0.829   0.4070
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 174.53  on 196  degrees of freedom
## AIC: 182.53
##
## Number of Fisher Scoring iterations: 4
```

```r
# sbp: SE > Estimate
0.01672/0.01386  # = SE 1.2 times > estimate
```

```
## [1] 1.206349
```

with the ratio of 1.2, it is resonable to choose `mlg_cad1: cad ~ dbp + gender` model.

2.4.2.5 Interaction, *

Interaction is the predictor variable combination that necessitates the interpretation of regression coefficients separately based for each level of the predictor (e.g. separate analysis for male vs female). Again, this makes interpreting our analysis complicated. So, most of the time, we pray not to have interaction in our regression model.

```r
summary(glm(cad ~ dbp*gender, data = coronary, family = binomial))
```

```
##
## Call:
## glm(formula = cad ~ dbp * gender, family = binomial, data = coronary)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.3876  -0.6677  -0.5317  -0.3306   2.4107
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -7.06999    2.50172  -2.826  0.00471 **
## dbp            0.06029    0.02807   2.148  0.03169 *
## genderman      2.11719    2.91088   0.727  0.46702
## dbp:genderman -0.01501    0.03288  -0.456  0.64815
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 174.99  on 196  degrees of freedom
## AIC: 182.99
##
## Number of Fisher Scoring iterations: 5
```

```
# insig. dbp*gender
```

There was no significant interaction to be included in out model.

2.4.3 Model fit assessment

There are three model fit assessment methods commonly done for logistic regression:

1. Hosmer-Lemeshow test.
2. Classification table.
3. Area Under the Curve (AUC) of Receiver Operating Characteristics (ROC) curve.

Basically, we want to compare the real cad status (observed) against the predicted cad status and probability (as predicted by our logistic regression model).

1. Hosmer-Lemeshow test.

    - P-value > 0.05 – Model (predicted counts) fit the data (observed counts).

```
# install.packages("ResourceSelection")
library(ResourceSelection)
hl_cad1 = hoslem.test(mlg_cad1$y, mlg_cad1$fitted.values)
hl_cad1  # does not fit
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mlg_cad1$y, mlg_cad1$fitted.values
## X-squared = 18.199, df = 8, p-value = 0.01978
```

P-value < 0.05, the model does not fit (slightly). Ideally > 0.05. Usually this happens because of small number of variables in the model.

Detailed counts,

```
cbind(hl_cad1$observed, hl_cad1$expected)
```

```
##                  y0 y1     yhat0    yhat1
## [0.0374,0.0657] 20  2 20.711530 1.288470
## (0.0657,0.0875] 18  2 18.368872 1.631128
## (0.0875,0.123]  22  0 19.644094 2.355906
## (0.123,0.136]   24  0 20.787142 3.212858
## (0.136,0.159]   11  2 11.005310 1.994690
## (0.159,0.18]    16  3 15.748367 3.251633
## (0.18,0.205]    14 10 19.208277 4.791723
## (0.205,0.239]   15  3 13.872019 4.127981
## (0.239,0.319]   11  9 14.170991 5.829009
## (0.319,0.652]   12  6  9.483399 8.516601
```

2. Classification table.

    - Cross-tabulate cad observed cad status vs predicted cad status.
    - Good model fit if $> 70\%$ of the subjects are correctly classified.

We must create probability and predicted cad variables, `cad_prob` and `cad_pred`,

```
coronary$cad_prob = mlg_cad1$fitted.values  # probability of cad from our model
head(coronary[c("cad", "cad_prob")])
```

```
##      cad   cad_prob
## 1 no cad 0.05985186
## 2 no cad 0.09456561
## 3 no cad 0.12324054
## 4 no cad 0.23685057
## 5 no cad 0.24845622
## 6 no cad 0.14799425
```

We set cutoff of probability (`cad_prob`) ≤ 0.5 for `no cad` and probability > 0.5 for `cad`,

```
coronary$cad_pred = cut(coronary$cad_prob, breaks = c(-Inf, 0.5, Inf),
                        labels = c("no cad", "cad"))  # the predicted cad status
head(coronary[c("cad", "cad_prob", "cad_pred")])
```

```
##      cad   cad_prob cad_pred
## 1 no cad 0.05985186   no cad
## 2 no cad 0.09456561   no cad
## 3 no cad 0.12324054   no cad
## 4 no cad 0.23685057   no cad
## 5 no cad 0.24845622   no cad
## 6 no cad 0.14799425   no cad
```

Cross-tabulate `cad` vs `cad_predicted`,

```
table(coronary$cad, coronary$cad_pred)
```

```
##
##           no cad cad
##   no cad     157   6
##   cad         34   3
```

Then calculate the correctly classified %,

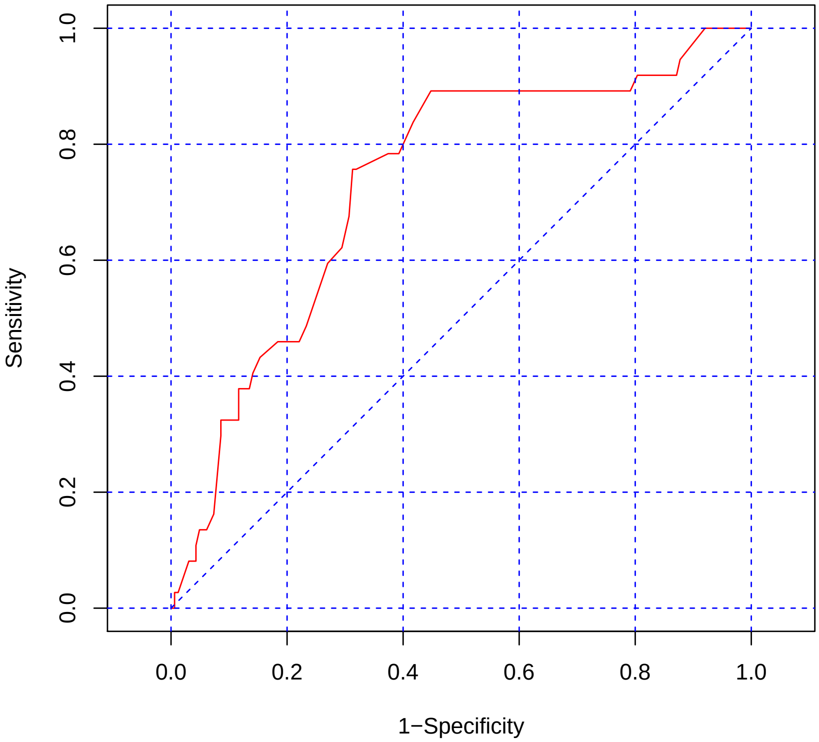
```
# correctly classified %
100 * (157 + 3) / length(coronary$cad)  # = 80%
```

```
## [1] 80
```

3. Area Under the Curve (AUC) of Receiver Operating Characteristics (ROC) curve.

- It measures the ability of a model to disciminate cad vs non-cad subjects.
- AUC is also known as C-statistic ("C" stands for "concordance").
- AUC $>$ 0.7 indicates acceptable model fit.
- AUC $\leq$ 0.5 shows no discrimination at all, unaceptable.

```
roc_cad1 = lroc(mlg_cad1)
```



```
roc_cad1$auc  # acceptable
```

```
## [1] 0.7320511
```

The model fulfill 2 out of 3 criteria we set for model fit assessment.

2.4.4 Interpretation

Now we have decided on our final model, rename the model,

```
# rename the selected model
mlg_cad_final = mlg_cad1
```

and interpret the ORs of the model,

```
summary(mlg_cad_final)
```

```
##
## Call:
## glm(formula = cad ~ dbp + gender, family = binomial, data = coronary)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4520  -0.6508  -0.5249  -0.3643   2.3337
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.12046    1.31667  -4.648 3.34e-06 ***
## dbp          0.04950    0.01463   3.383 0.000717 ***
## genderman    0.80573    0.39084   2.062 0.039253 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 191.56  on 199  degrees of freedom
## Residual deviance: 175.20  on 197  degrees of freedom
## AIC: 181.2
##
## Number of Fisher Scoring iterations: 4
```

```
exp(Confint(mlg_cad_final))  # ORs and the 95% CIs
```

```
##                Estimate        2.5 %     97.5 %
## (Intercept) 0.002197438 0.0001460685 0.02632238
## dbp         1.050750205 1.0217691211 1.08250612
## genderman   2.238324210 1.0552821023 4.93500645
```

- 1mmHg increase in DBP increase the odds of cad by 1.05 times (or 5%), controlling the effect of gender.
- Man has 2.24 times odds of cad as compared to woman, controlling for the effect of DBP.

Notice that for numerical predictor, it sounds odd to interpret the OR for 1 unit increase. We can obtain the OR for any specific increase in the value (a constant, c), e.g. 5 or 10 unit increase etc. To obtain the OR simply multiply the coefficient *beta* (careful, not OR) by the needed constant value, c\$,

$$OR = e^{(c \times \beta)}$$

To obtain the OR of 10mmHg increase in DBP,

$$OR_{10 \times dbp} = e^{10 \times 0.05} = e^{0.5} = 1.65$$

```
exp(10*0.05)
```

```
## [1] 1.648721
```

or more precisely, directly from our model,

```
coef(mlg_cad_final)
```

```
## (Intercept)         dbp   genderman
## -6.12046337  0.04950439  0.80572747
```

```
exp(10*coef(mlg_cad_final)[2])
```

```
##     dbp
## 1.64057
```

- 10mmHg increase in DBP increase the odds of cad by 1.64 times (or 64%), controlling the effect of gender.

We can also obtain R^2 for the logistic regression model,

```
rsq(mlg_cad_final, adj = T)
```

```
## [1] 0.07257276
```

- DBP and gender explains (only) 7.3% variance in cad. This is quite low, which indicates that there are more predictors we should consider to predict cad occurrence.

Note: R-squared is usually reported for linear regression. But R-squared is also available for GLM, in our case logistic regression. This is usually known as pseudo-R-squared. In GLM, it is made possible by the work of Zhang (2017), the author of "rsq" package.

2.4.5 Model equations

Our basic logistic regression equation is given by,

$$log_e\left(\frac{p_{cad}}{1-p_{cad}}\right) = -6.12 + 0.05 \times \ dbp + 0.81 \times gender \ (man)$$

CAD probability is given by,

$$p_{cad} = \frac{e^{-6.12+0.05\times \ dbp+0.81\times gender \ (man)}}{1+e^{-6.12+0.05\times \ dbp+0.81\times gender \ (man)}}$$

Note: Again, don't scratch your head.

2.4.6 Prediction

It is easy to predict in R using our fitted model above. First we view the predicted values for our sample,

```
coronary$cad_prob1 = predict(mlg_cad_final, type = "response")  # in probability
# converted from logit, by adding type = "response"
head(coronary)
```

```
##      cad sbp dbp   chol age  bmi   race gender   cad_prob cad_pred
## 1 no cad 106  68 6.5725  60 38.9 indian  woman 0.05985186   no cad
## 2 no cad 130  78 6.3250  34 37.8  malay  woman 0.09456561   no cad
## 3 no cad 136  84 5.9675  36 40.5  malay  woman 0.12324054   no cad
## 4 no cad 138 100 7.0400  45 37.6  malay  woman 0.23685057   no cad
## 5 no cad 115  85 6.6550  53 40.3 indian    man 0.24845622   no cad
## 6 no cad 124  72 5.9675  43 37.6  malay    man 0.14799425   no cad
##    cad_prob1
## 1 0.05985186
## 2 0.09456561
## 3 0.12324054
## 4 0.23685057
## 5 0.24845622
## 6 0.14799425
```

You can also use `mlg_cad_final$fitted.values` as we did before for `cad_prob`. But as we will see below, we need predict() for new data, so we need to use the proper `predict()` function.

Now let us try predicting for some new values,

```
str(coronary[c("dbp", "gender")])
```

```
## 'data.frame':    200 obs. of  2 variables:
##  $ dbp   : num  68 78 84 100 85 72 80 70 85 70 ...
##  $ gender: Factor w/ 2 levels "woman","man": 1 1 1 1 2 2 2 1 1 2 ...
```

```
# simple, dbp = 110, gender = man
predict(mlg_cad_final, list(dbp = 110, gender = "man"), type = "response")
```

```
##         1
## 0.5326403
```

```
# probability > 0.5 = cad
```

More data points,

```
new_data = data.frame(dbp = c(100, 110, 120, 100, 110, 120),
                      gender = c("man", "man", "man", "woman", "woman", "woman"))
new_data
```

```
##   dbp gender
## 1 100    man
## 2 110    man
## 3 120    man
## 4 100  woman
## 5 110  woman
## 6 120  woman
```

```
predict(mlg_cad_final, new_data, type = "response")
```

```
##         1         2         3         4         5         6
## 0.4099198 0.5326403 0.6515344 0.2368506 0.3373825 0.4551368
```

```
new_data$cad_prob = predict(mlg_cad_final, new_data, type = "response")
new_data
```

```
##   dbp gender  cad_prob
## 1 100    man 0.4099198
## 2 110    man 0.5326403
## 3 120    man 0.6515344
## 4 100  woman 0.2368506
## 5 110  woman 0.3373825
## 6 120  woman 0.4551368
```

```
new_data$cad_pred = cut(new_data$cad_prob, breaks = c(-Inf, 0.5, Inf),
                        labels = c("no cad", "cad"))
new_data
```

```
##   dbp gender  cad_prob cad_pred
## 1 100    man 0.4099198   no cad
## 2 110    man 0.5326403      cad
## 3 120    man 0.6515344      cad
## 4 100  woman 0.2368506   no cad
## 5 110  woman 0.3373825   no cad
## 6 120  woman 0.4551368   no cad
```

2.5 Exercises

1. Present the results in a table (follow Arifin et al. (2016))

2. Obtain the OR for 5mmHg increase in DBP.
3. Repeat the analysis using “coronary_large.sav” dataset.

References

Arifin, W. N., Sarimah, A., Norsa'adah, B., Majdi, Y. N., Siti-Azrin, A. H., Imran, M. K., … Naing, L. (2016). Reporting statistical results in medical journals. *The Malaysian Journal of Medical Sciences: MJMS*, *23*(5), 1.

Chongsuvivatwong, V. (2018). *EpiDisplay: Epidemiological data display package*. Retrieved from https://CRAN.R-project.org/package=epiDisplay

Fox, J., Weisberg, S., & Price, B. (2018). *Car: Companion to applied regression*. Retrieved from https://CRAN.R-project.org/package=car

Hosmer, D., Lemeshow, S., & Sturdivant, R. (2013). *Applied logistic regression*. Wiley. Retrieved from https://books.google.com.my/books?id=bRoxQBIZRd4C

Lele, S. R., Keim, J. L., & Solymos, P. (2017). *ResourceSelection: Resource selection (probability) functions for use-availability data*. Retrieved from https://CRAN.R-project.org/package=ResourceSelection

R Core Team. (2018a). *Foreign: Read data stored by 'minitab', 's', 'sas', 'spss', 'stata', 'systat', 'weka', 'dBase', …* Retrieved from https://CRAN.R-project.org/package=foreign

R Core Team. (2018b). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Revelle, W. (2018). *Psych: Procedures for psychological, psychometric, and personality research*. Retrieved from https://CRAN.R-project.org/package=psych

Ripley, B. (2018). *MASS: Support functions and datasets for venables and ripley's mass*. Retrieved from https://CRAN.R-project.org/package=MASS

Sarkar, D. (2017). *Lattice: Trellis graphics for r*. Retrieved from https://CRAN.R-project.org/package=lattice

Zhang, D. (2017). A coefficient of determination for generalized linear models. *The American Statistician*, *71*(4), 310–316. https://doi.org/10.1080/00031305.2016.1256839

Zhang, D. (2018). *Rsq: R-squared and related measures*. Retrieved from https://CRAN.R-project.org/package=rsq