Exploratory and Inferential Data analysis

Wan Nor Arifin

Unit of Biostatistics and Research Methodology, Universiti Sains Malaysia.

email: wnarifin@usm.my

December 19, 2018

Preliminaries

Library

```
# library
library(foreign)
library(psych)
library(epiDisplay)
library(rsq)
library(car)
library(survival)
```

Data

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

Descriptive statistics

Basic

```
# basic
summary(coronary)
```

Numerical

```
# numerical
summ(subset(coronary, select = c(sbp, dbp, chol, age, bmi)))
describe(subset(coronary, select = c(sbp, dbp, chol,
                                     age, bmi)))
mapply(IQR, subset(coronary, select = c(sbp, dbp, chol,
                                        age, bmi)))
# replace IQR with mean, median etc functions
```

Categorical

```
# categorical
codebook(subset(coronary, select = c(cad, race, gender)))
codebook(coronary)  # can handle both numerical
# & categorical
```

Statistics by group

```
# stats by group
by(coronary, coronary$cad, summary)
describeBy(subset(coronary,
                  select = c(sbp, dbp, chol, age, bmi)),
           coronary$cad)
by(subset(coronary, select = c(race, gender)),
   coronary$cad, table)
by(subset(coronary, select = c(race, gender)),
   coronary$cad, codebook)
```

Cross tabulation

```
# cross tabulation
table(coronary$cad, coronary$gender)
table(coronary$cad, coronary$race)
```

Linear regression

Data

```
# data
data_lr = subset(coronary, select = c(chol, dbp, race))
str(data_lr)
```

Descriptive

```
# descriptive
codebook(data_lr)
plot(data_lr)
```

Multiple linear regression

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

Multiple linear regression

```
# mlr model, chol ~ dbp + race
mlr_chol = glm(chol ~ dbp + race, data = data_lr)
summary(mlr_chol)
Confint(mlr_chol)  # 95% CI of the coefficients
rsq(mlr_chol, adj = T)

regress.display(mlr_chol)  # epiDisplay
```

Prediction

```
# predict
data_lr$pred_chol = predict(mlr_chol)
head(data_lr)
# simple, dbp = 90, race = indian
predict(mlr_chol, list(dbp = 90, race = "indian"))
```

Prediction

```
# more data points
new_data = data.frame(dbp = c(90, 90, 90),
                      race = c("malay", "chinese",
                               "indian"))
new_data
new_data$pred_chol = predict(mlr_chol, new_data)
new_data
```

Logistic regression

Data

```
# data
data_logr = subset(coronary, select = c(cad, dbp, gender))
str(data_logr)
```

Descriptive

```
# descriptive, by CAD
codebook(data_logr)
by(subset(data_logr, select = c(dbp, gender)),
   data_logr$cad, codebook)
```

Multiple logistic regression

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

Multiple logistic regression

More accurately,

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

Multiple logistic regression

```
# mlogr, log(cad odds) ~ dbp + gender
mlg_cad = glm(cad ~ dbp + gender, data = coronary,
              family = binomial)
summary(mlg_cad)
Confint(mlg_cad)  # 95% CI of the coefficients
exp(Confint(mlg_cad))  # ORs and the 95% CIs
rsq(mlg_cad, adj = T)  # pseudo R-squared

logistic.display(mlg_cad)  # epiDisplay
```

Multiple logistic regression

```
# model fit
lroc(mlg_cad)  # ROC
lroc(mlg_cad)$auc  # AUC
library(ResourceSelection)  # Hosmer-Lemeshow test
hoslem.test(mlg_cad$y, mlg_cad$fitted.values)
```

Prediction

```
# predict
data_logr$cad_prob = predict(mlg_cad, type = "response")
# in probability
# converted from logit, by adding type = "response"
head(data_logr)
# simple, dbp = 110, gender = man
predict(mlg_cad, list(dbp = 110, gender = "man"),
        type = "response")
# probability > 0.5 = cad
```

Prediction

```
# 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
new_data$prob_cad = predict(mlg_cad, new_data,
                            type = "response")
new_data
new_data$pred_cad = cut(new_data$prob_cad,
                        breaks = c(-Inf, 0.5, Inf),
                        labels = c("no cad", "cad"))
new_data
```

Cox regression

Data

```
# data
lca = subset(lung, select = c(status, time, sex))
# from survival package
str(lca)
table(lca$status)  # status: 1=censored, 2=dead
lca$status = lca$status - 1  # turn to our usual 0/1
lca$sex = factor(lca$sex, levels = 1:2,
                 labels = c("male", "female"))
str(lca)
```

Descriptive

```
# descriptive
codebook(lca)
table(lca$status)   # number of events
```

Cox regression

Cox proportional hazards (PH) model,

$$log_e\left(\frac{hazard\ at\ time,\ t}{baseline\ hazard\ at\ time,\ t}\right) =$$

$$log_e(hazard\ ratio,\ HR) = coefficients \times numerical\ predictors + coefficients \times categorical\ predictors$$

Cox regression

```
# coxr, log(hazard ratio) ~ sex
cox_lca = coxph(Surv(time, status) ~ sex, data = lca)
summary(cox_lca)
cox.zph(cox_lca)  # Prop. hazards assumption
# -- test constant coefficients over time
```

Prediction

```
# predict
# obtain hazard/risk
lca$hazard = predict(cox_lca, type = "risk")
lca
```

Prediction

```
# obtain median survival time & probability, sex = "female"
new_data = data.frame(sex = c("male", "female"))
new_data
new_cox = survfit(cox_lca, newdata = new_data)
new_cox  # median survival time
summary(new_cox, times = c(100, 200, 300))
# survival at 100, 200 and 300 days
```

Broom

Broom

The broom package takes the messy output of built-in functions in R, such as lm, nls, or t.test, and turns them into tidy data frames.

More information @
https://cran.r-project.org/web/packages/broom/vignettes/broom.html

```
library(broom)
```

Linear regression

```
# lr
tidy(mlr_chol)
augment(mlr_chol)
glance(mlr_chol)
```

Logistic regression

```
# logr
tidy(mlg_cad)
augment(mlg_cad)
glance(mlg_cad)
```

Cox regression

```
# coxr
tidy(cox_lca)
augment(cox_lca, lca)
glance(cox_lca)
```

Thank you

References

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

R Core Team. (2018). *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

Therneau, T. M. (2018). *Survival: Survival analysis*. Retrieved from https://CRAN.R-project.org/package=survival

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