R 語言與機器學習 (六)

丘祐瑋
David Chiu

是否可以預測H7N9的死亡率

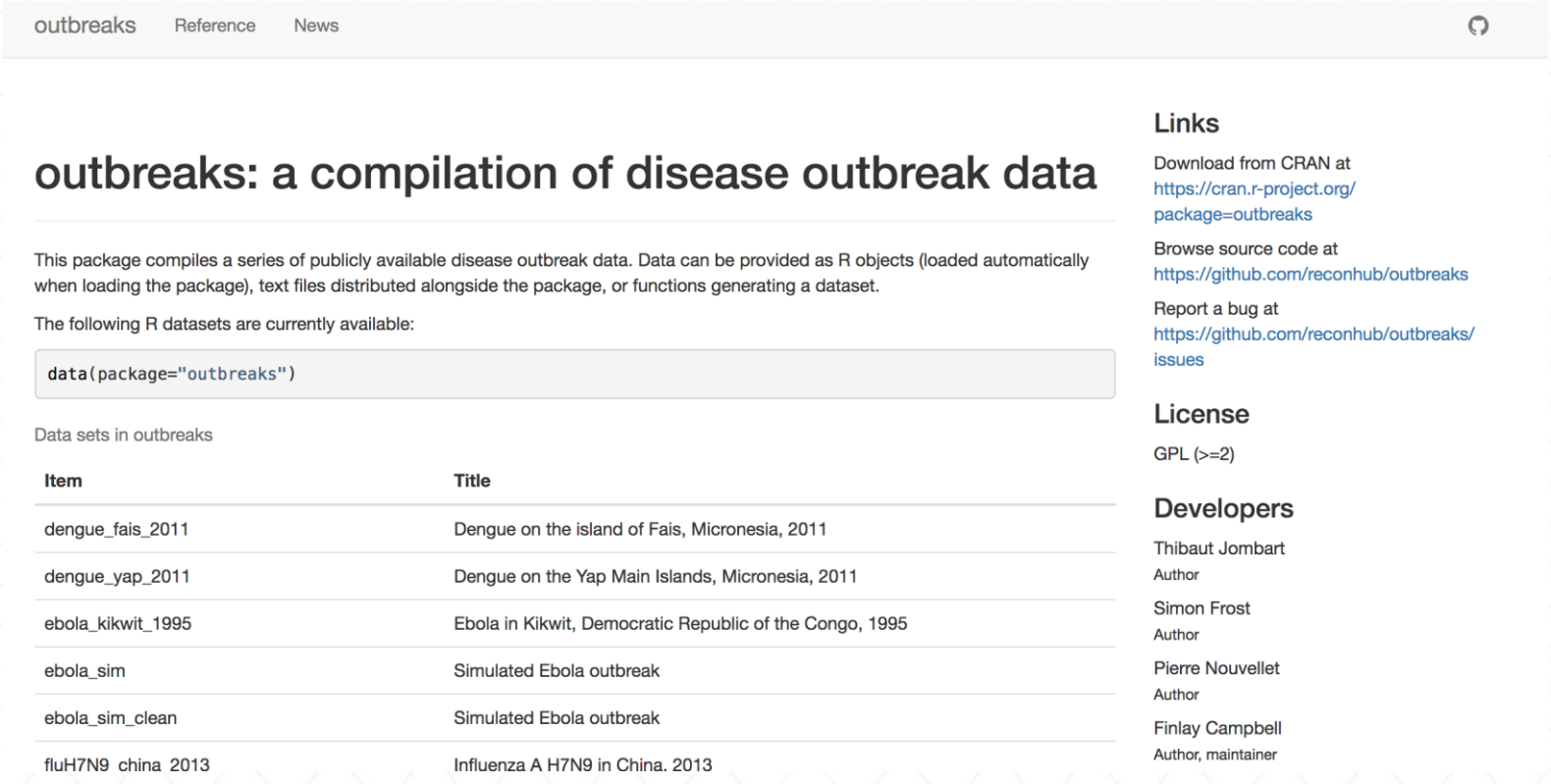
H7N9禽流感

- H7N9禽流感於2013年3月份在上海、安徽兩地出現了全球首次人類感染的案例，並隨後在2013年4月、2013年5月10個省市、39個地市相繼報告出現H7N9疫情，速度之快令人有些措手不及。

- 截至2013年5月31日，中國報告確診病例131例，死亡39人，死亡率30%，重症率近八成。這一數字遠高於2003年SAAS肆虐中國時7%的死亡率和3成重症率。截至2013年12月31日，中國內地共報告確診病例144例，死亡45人。

Outbreaks 資料集

- http://www.repidemicsconsortium.org/outbreaks/



outbreaks Reference News

outbreaks: a compilation of disease outbreak data

This package compiles a series of publicly available disease outbreak data. Data can be provided as R objects (loaded automatically when loading the package), text files distributed alongside the package, or functions generating a dataset.

The following R datasets are currently available:

```
data(package="outbreaks")
```

Data sets in outbreaks

Item	Title
dengue_fais_2011	Dengue on the island of Fais, Micronesia, 2011
dengue_yap_2011	Dengue on the Yap Main Islands, Micronesia, 2011
ebola_kikwit_1995	Ebola in Kikwit, Democratic Republic of the Congo, 1995
ebola_sim	Simulated Ebola outbreak
ebola_sim_clean	Simulated Ebola outbreak
fluH7N9_china_2013	Influenza A H7N9 in China. 2013

Links

Download from CRAN at https://cran.r-project.org/package=outbreaks

Browse source code at https://github.com/reconhub/outbreaks

Report a bug at https://github.com/reconhub/outbreaks/issues

License

GPL (>=2)

Developers

Thibaut Jombart
Author

Simon Frost
Author

Pierre Nouvellet
Author

Finlay Campbell
Author, maintainer

讀取中國2013年H7N9資料集

■ 安裝outbreaks 套件

```
install.packages("outbreaks")
```

■ 載入outbreaks 套件

```
library(outbreaks)
```

■ 讀取中國2013年H7N9資料集

```
data(fluH7N9_china_2013)
```

探索資料

檢視資料

```
# 檢視資料型態
class(fluH7N9_china_2013)

# 檢視資料概要
str(fluH7N9_china_2013)

# 檢視前幾筆資料
head(fluH7N9_china_2013)
```

case_id	案例編號
date_of_onset	症狀發作日期
date_of_hospitalisation	住院日期
date_of_outcome	治療結果日期
outcome	治療結果
gender	性別
age	年紀
province	省分

資料前處理

```
# 將 ? 轉換為 NA
fluH7N9_china_2013$age[which(fluH7N9_china_2013$age == "?")] <- NA

# 將 age 轉換為數值型態
fluH7N9_china_2013$age <- as.numeric(fluH7N9_china_2013$age )

# 新增 case ID 資料
fluH7N9_china_2013$case_id  <- paste("case", fluH7N9_china_2013$case_id,
sep = "_")
```

使用tidyr 轉換資料

```
library(tidyr)
fluH7N9_china_2013_gather <- fluH7N9_china_2013 %>%
  gather(Group, Date, date_of_onset:date_of_outcome)

fluH7N9_china_2013_gather
```

outcome <fctr>	gender <fctr>	age <dbl>	province <fctr>	Group <chr>	Date <date>
Death	m	58	Shanghai	date_of_onset	2013-02-19
Death	m	7	Shanghai	date_of_onset	2013-02-27
Death	f	11	Anhui	date_of_onset	2013-03-09
NA	f	18	Jiangsu	date_of_onset	2013-03-19
Recover	f	20	Jiangsu	date_of_onset	2013-03-19
Death	f	9	Jiangsu	date_of_onset	

根據住院,治療, 結果日期分組表列資料

資料轉換

```
# 重新編排資料順序
fluH7N9_china_2013_gather$Group <- factor(fluH7N9_china_2013_gather$Group, levels =
c("date_of_onset", "date_of_hospitalisation", "date_of_outcome"))

# 重新命名群組
library(plyr)
fluH7N9_china_2013_gather$Group <-mapvalues(fluH7N9_china_2013_gather$Group, from =
c("date_of_onset", "date_of_hospitalisation", "date_of_outcome"),
        to = c("Date of onset", "Date of hospitalisation", "Date of outcome"))

# 將江蘇,上海與浙江以外的地區列為其他
fluH7N9_china_2013_gather$province <- mapvalues(fluH7N9_china_2013_gather$province,from =
c("Anhui", "Beijing", "Fujian", "Guangdong", "Hebei", "Henan", "Hunan", "Jiangxi", "Shandong", "Taiwan"),
to = rep("Other", 10))
```

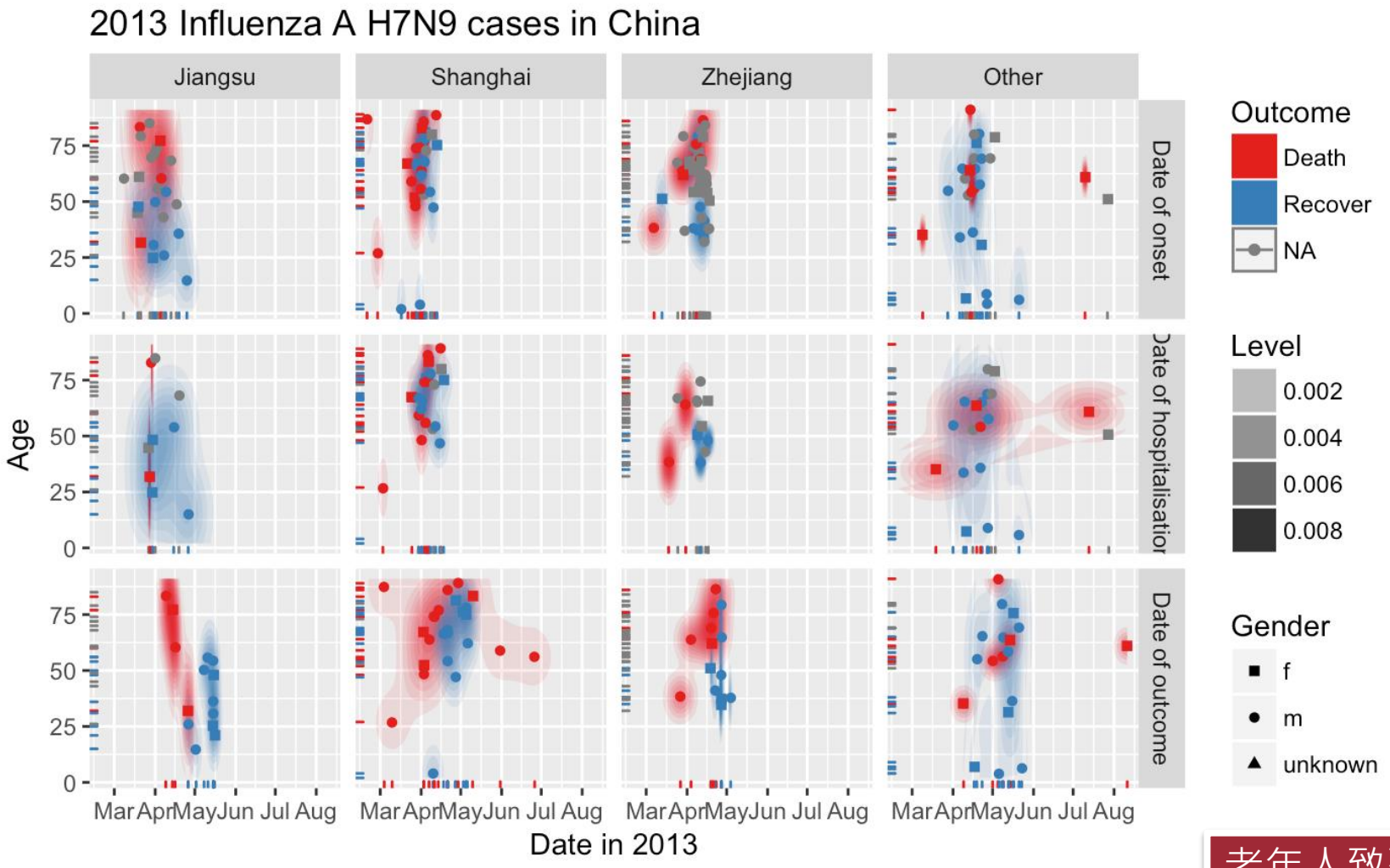
資料轉換

```
# 為未知性別增添類別
levels(fluH7N9_china_2013_gather$gender) <- c(levels(fluH7N9_china_2013_gather$gender), "unknown")

fluH7N9_china_2013_gather$gender[is.na(fluH7N9_china_2013_gather$gender)] <- "unknown"

# 只列出江蘇, 上海, 浙江及其他地區等省分資訊
fluH7N9_china_2013_gather$province <- factor(fluH7N9_china_2013_gather$province, levels = c("Jiangsu",  "Shanghai", "Zhejiang", "Other"))
```

繪製資料

```
ggplot(data = fluH7N9_china_2013_gather, aes(x = Date, y = age, fill = outcome)) +
  stat_density2d(aes(alpha = ..level..), geom = "polygon") +
  geom_jitter(aes(color = outcome, shape = gender), size = 1.5) +
  geom_rug(aes(color = outcome)) +
  labs(
    fill = "Outcome",
    color = "Outcome",
    alpha = "Level",
    shape = "Gender",
    x = "Date in 2013",
    y = "Age",
    title = "2013 Influenza A H7N9 cases in China"
  ) +
  facet_grid(Group ~ province) +
  scale_shape_manual(values = c(15, 16, 17)) +
  scale_color_brewer(palette="Set1", na.value = "grey50") +
  scale_fill_brewer(palette="Set1")
```

資料視覺化

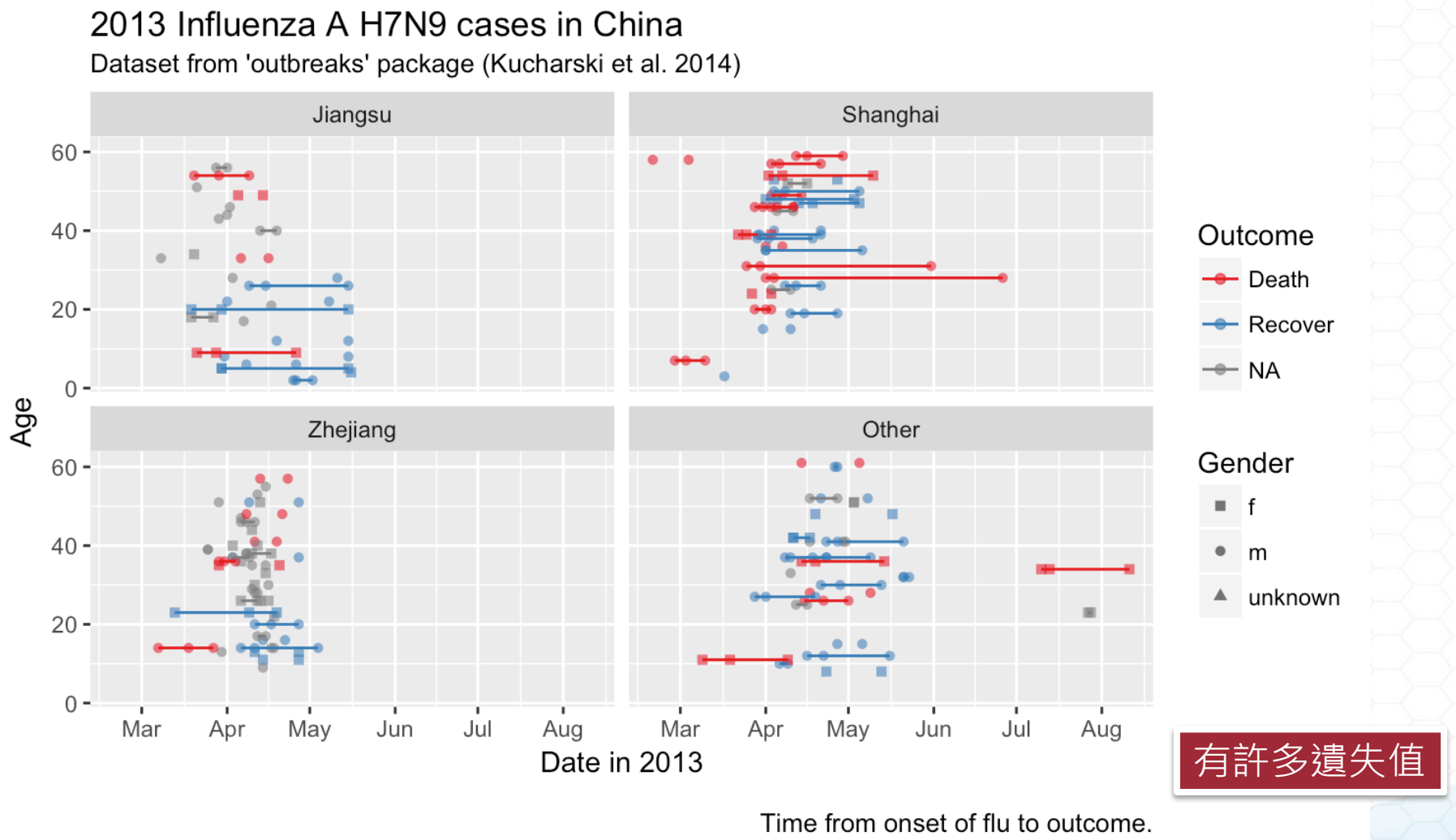


老年人致死率比較高

觀察治療時間長短與死亡率的關係

```
ggplot(data = fluH7N9_china_2013_gather, aes(x = Date, y = age, color = outcome)) +
  geom_point(aes(color = outcome, shape = gender), size = 1.5, alpha = 0.6) +
  geom_path(aes(group = case_id)) +
  facet_wrap( ~ province, ncol = 2) +
  scale_shape_manual(values = c(15, 16, 17)) +
  scale_color_brewer(palette="Set1", na.value = "grey50") +
  scale_fill_brewer(palette="Set1") +
  labs(
    color = "Outcome",
    shape = "Gender",
    x = "Date in 2013",
    y = "Age",
    title = "Time from onset of flu to outcome"
  )
```

觀察治療時間長短與死亡率的關係



有許多遺失值

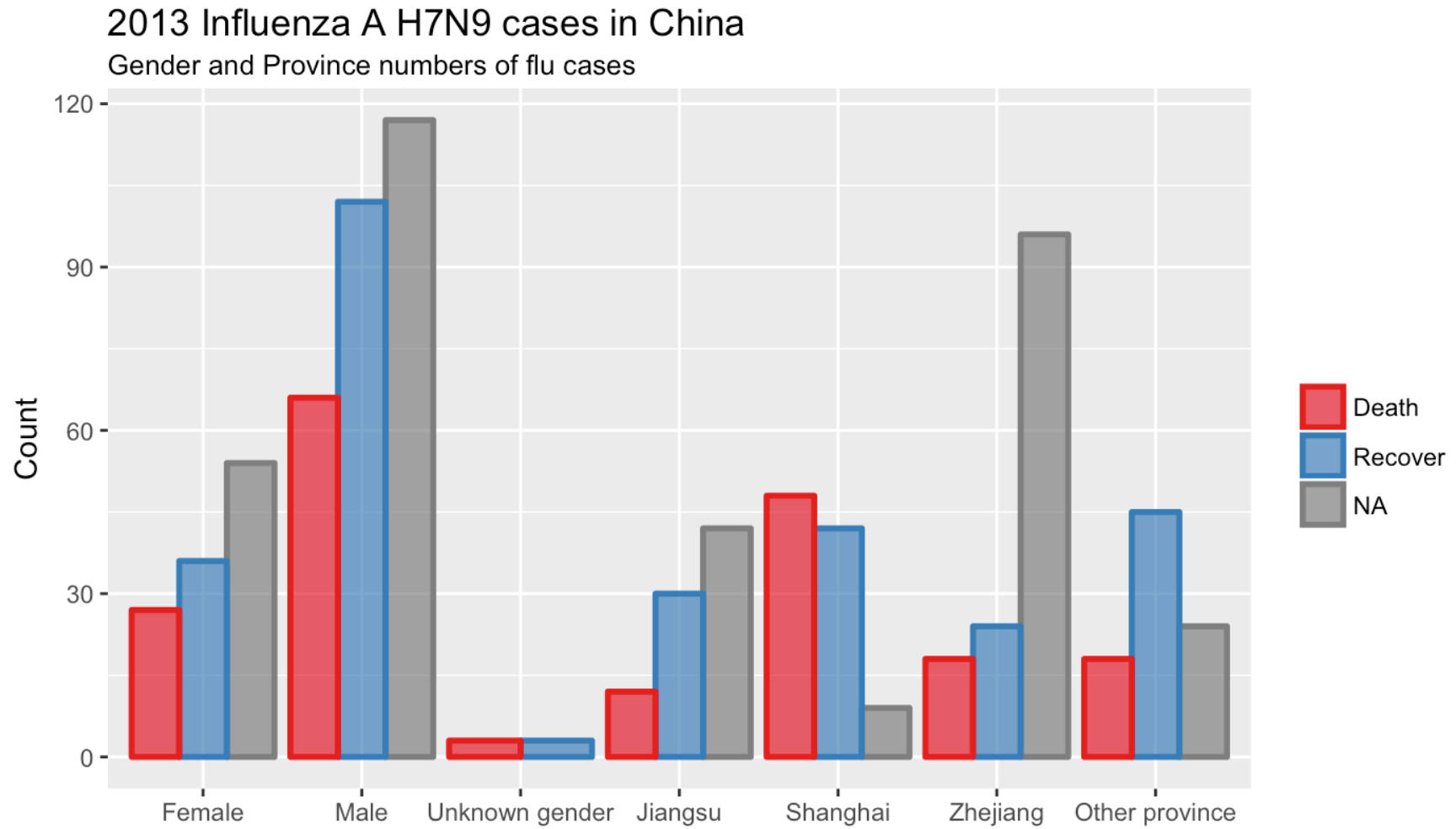
資料轉換

```
fluH7N9_china_2013_gather_2 <- fluH7N9_china_2013_gather[, -4] %>%
  gather(group_2, value, gender:province)

fluH7N9_china_2013_gather_2$value <-
mapvalues(fluH7N9_china_2013_gather_2$value, from = c("m", "f", "unknown",
"Other"), to = c("Male", "Female", "Unknown gender", "Other province"))

fluH7N9_china_2013_gather_2$value <-
factor(fluH7N9_china_2013_gather_2$value, levels = c("Female", "Male",
"Unknown gender", "Jiangsu", "Shanghai", "Zhejiang", "Other province"))
```

根據性別與區域繪製病例長條圖

```
p1 <- ggplot(data = fluH7N9_china_2013_gather_2, aes(x = value, fill = outcome, color = outcome)) +
  geom_bar(position = "dodge", alpha = 0.7, size = 1) +
  scale_fill_brewer(palette="Set1", na.value = "grey50") +
  scale_color_brewer(palette="Set1", na.value = "grey50") +
  labs(
    color = "",
    fill = "",
    x = "",
    y = "Count",
    title = "2013 Influenza A H7N9 cases in China")
```

根據性別與區域繪製病例長條圖



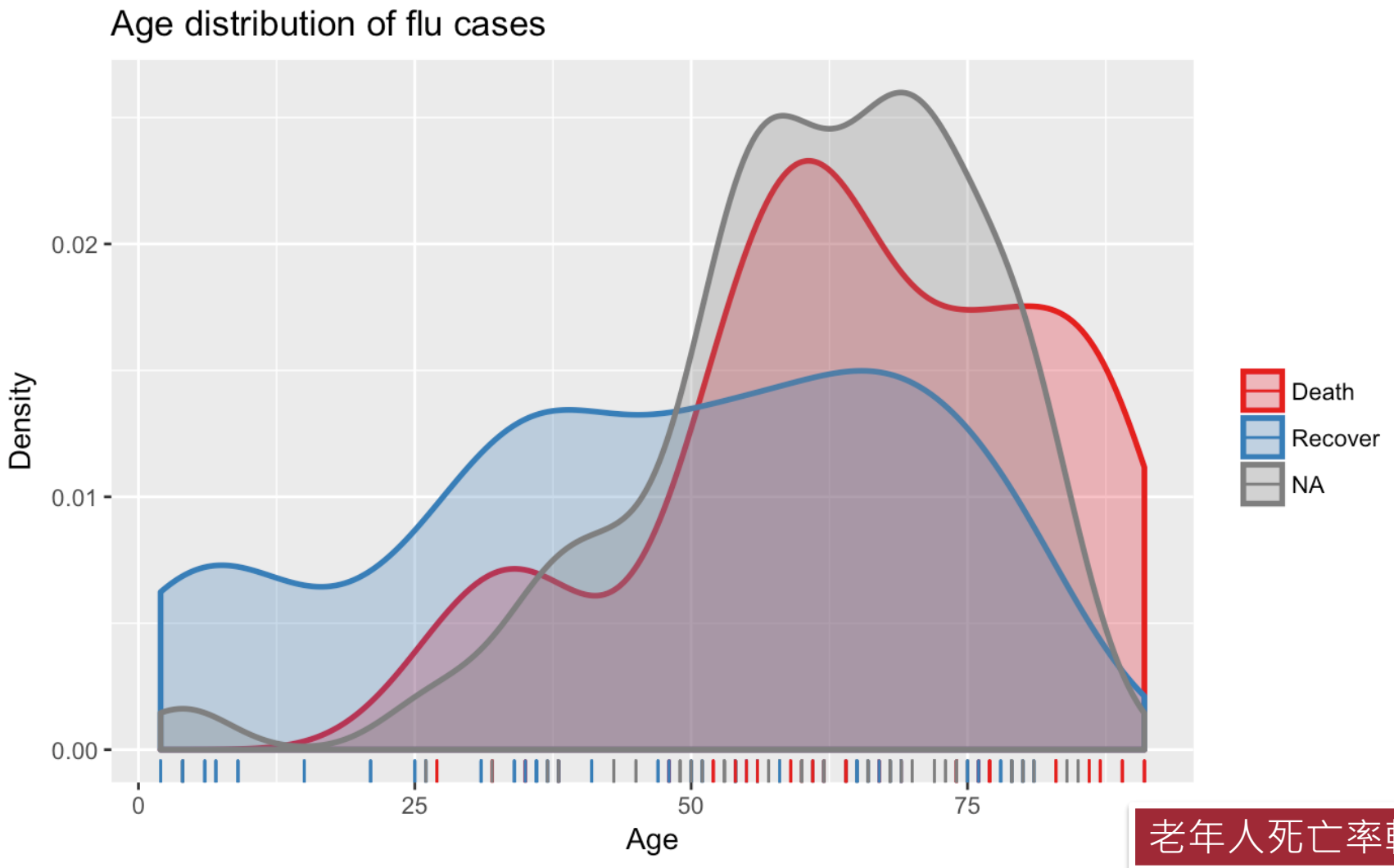
男女死亡率差不多

上海死亡率較高

根據年紀繪製流感案例分布

```
p2 <- ggplot(data = fluH7N9_china_2013_gather, aes(x = age, fill = outcome, color = outcome)) +
  geom_density(alpha = 0.3, size = 1) +
  geom_rug() +
  scale_color_brewer(palette="Set1", na.value = "grey50") +
  scale_fill_brewer(palette="Set1", na.value = "grey50") +
  labs(
    color = "",
    fill = "",
    x = "Age",
    y = "Density",
    title = "Age distribution of flu cases"
  )
```

根據年紀繪製流感案例分布



老年人死亡率較高

抽取特徵

資料前處理

```
# 讀取中國2013年H7N9資料集
data(fluH7N9_china_2013)

# 建立 dataset
dataset <- fluH7N9_china_2013

# 將 ? 轉換為 NA, 將年紀轉換為數值
dataset $age[which(dataset$age == "?")] <- NA
dataset$age <-  as.numeric(as.character(dataset$age))

# 新增 case ID 資料
dataset$case_id <- paste("case", dataset$case_id, sep = "_")
```

產生虛擬變量

```
dataset$hospital <-  as.factor(ifelse(is.na(dataset$date_of_hospitalisation), 0, 1))
dataset$gender_f <- as.factor(ifelse(dataset$gender == "f", 1, 0))
dataset$province_Jiangsu <- as.factor(ifelse(dataset$province == "Jiangsu", 1, 0))
dataset$province_Shanghai <- as.factor(ifelse(dataset$province == "Shanghai", 1, 0))
dataset$province_Zhejiang <- as.factor(ifelse(dataset$province == "Zhejiang", 1, 0))
dataset$province_other   <- as.factor(ifelse(dataset$province == "Zhejiang" | dataset$province
== "Jiangsu" | dataset$province == "Shanghai", 0, 1))
```

建立欄位數為變量數 - 1

轉換日期資料

```
# 計算發作到治療完畢的日期
dataset$days_onset_to_outcome <- as.numeric(as.character(gsub(" days", "",
            as.Date(as.character(dataset$date_of_outcome), format = "%Y-%m-%d") -
             as.Date(as.character(dataset$date_of_onset), format = "%Y-%m-%d"))))

# 計算發作到住院日期
dataset$days_onset_to_hospital <- as.numeric(as.character(gsub(" days", "",
        as.Date(as.character(dataset$date_of_hospitalisation), format = "%Y-%m-%d") -
as.Date(as.character(dataset$date_of_onset), format = "%Y-%m-%d"))))

# 找出早期治療的病例
dataset$early_onset <- as.factor(ifelse(dataset$date_of_onset < summary(dataset$date_of_onset)[[3]], 1, 0))

# 找出早期有醫療結果的病例
dataset$early_outcome <- as.factor(ifelse(dataset$date_of_outcome < summary(dataset$date_of_outcome)[[3]], 1, 0))
```

取得需要特徵資料

```
dataset <- dataset[,c('case_id','outcome', 'age', 'hospital', 'gender_f', 'province_Jiangsu',
'province_Shanghai', 'province_Zhejiang', 'province_other', 'days_onset_to_outcome',
'days_onset_to_hospital', 'early_onset', 'early_outcome')]
```

	case_id	outcome	age	hospital	gender_f	province_Jiangsu	province_Shanghai	province_Zhejiang	province_other
1	case_1	Death	87	0	0	0	1	0	0
2	case_2	Death	27	1	0	0	1	0	0
3	case_3	Death	35	1	1	0	0	0	1
4	case_4	*NA*	45	1	1	1	0	0	0
5	case_5	Recover	48	1	1	1	0	0	0
6	case_6	Death	32	1	1	1	0	0	0
7	case_7	Death	83	1	0	1	0	0	0
8	case_8	Death	38	1	0	0	0	1	0
9	case_9	*NA*	67	1	0	0	0	1	0
10	case_10	Death	48	1	0	0	1	0	0
11	case_11	Death	64	1	0	0	0	1	0
12	case_12	Death	52	0	1	0	1	0	0

使用MICE 填補遺失值

```
library(mice)
dataset_impute <- mice(data = dataset[, -2],  print = FALSE)
dataset_impute
```

在R的MICE套件，會計算資料的分布，並且根據分布填補遺失值

重組資料集

```
library(dplyr)
datasets_complete <- right_join(dataset[, c(1, 2)],
                                complete(dataset_impute, "long"),
                                by = "case_id") %>% select(-.id)
```

垂直堆疊資料
1) .imp 代表填補值
2) .id 代表資料編號

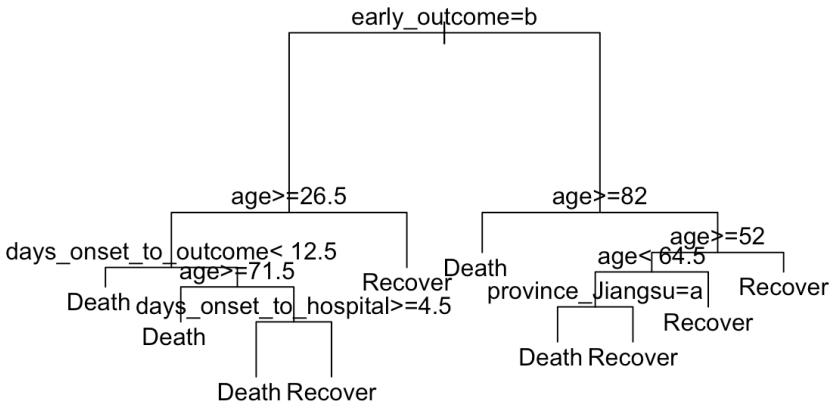
建立模型

準備訓練與測試資料

```
train_index <- which(is.na(datasets_complete$outcome))
train_data <- datasets_complete[-train_index, ]
test_data  <- datasets_complete[train_index, -2]

set.seed(42)
idx      <- sample.int(2 , nrow(train_data), p = c(0.7, 0.3), replace=TRUE)
val_train_data <- train_data[idx == 1, ]
val_test_data  <- train_data[idx == 2, ]
```

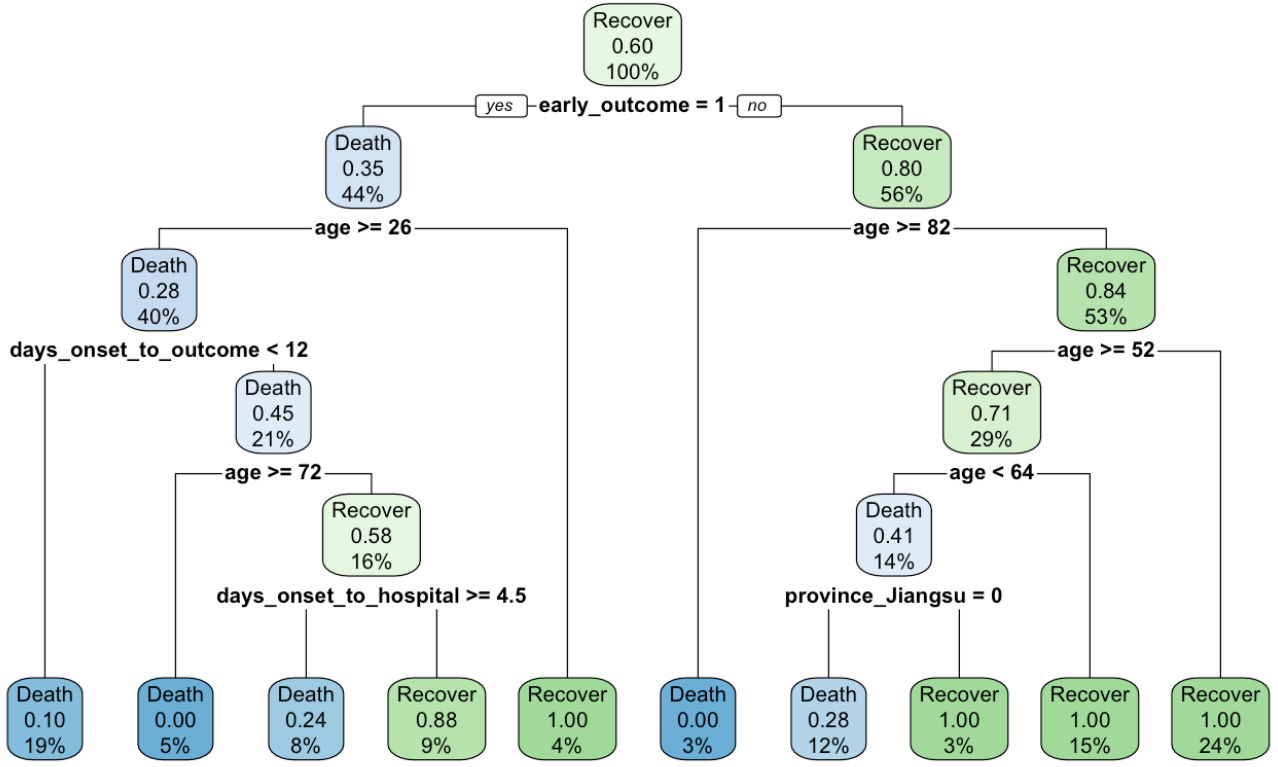
建立決策樹

```
library(rpart)
fit <- rpart(outcome ~., data = val_train_data[,-1])
plot(fit, margin = 0.1)
text(fit)
```



使用 rpart.plot 繪製決策樹

```
library(rpart.plot)
rpart.plot(fit)
```



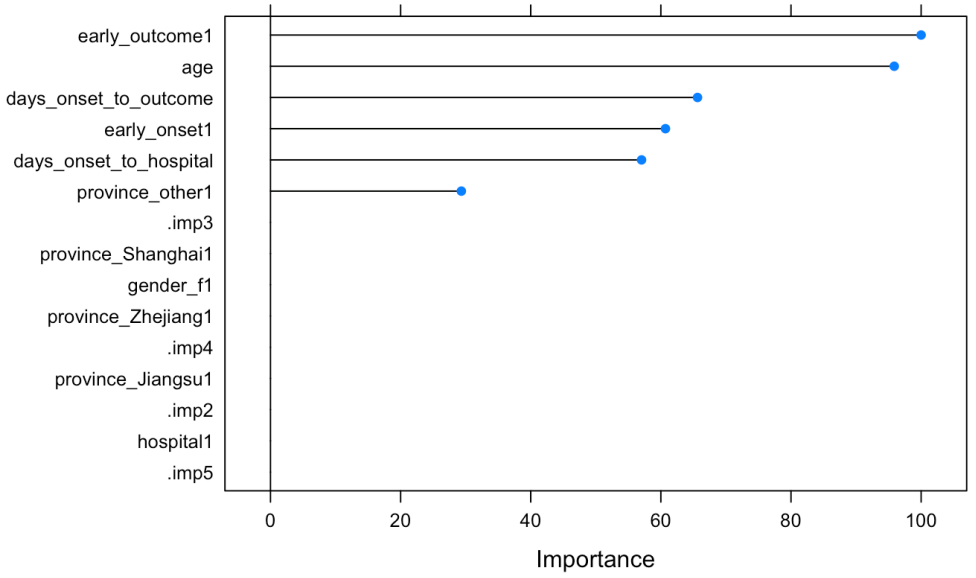
檢視模型準確度

```
predicted <- predict(fit, val_test_data, type = 'class')
table(val_test_data$outcome, predicted)
```

	predicted	
	Death	Recover
Death	49	2
Recover	10	61

評估特徵重要性

```
control <- trainControl(method = "repeatedcv", number = 10, repeats = 3)

set.seed(42)
model1 <- train(outcome ~ ., data = val_train_data[,-1], method = "rpart",
preProcess = NULL, trControl = control)

importance1 <- varImp(model1, scale=TRUE)
plot(importance1)
```

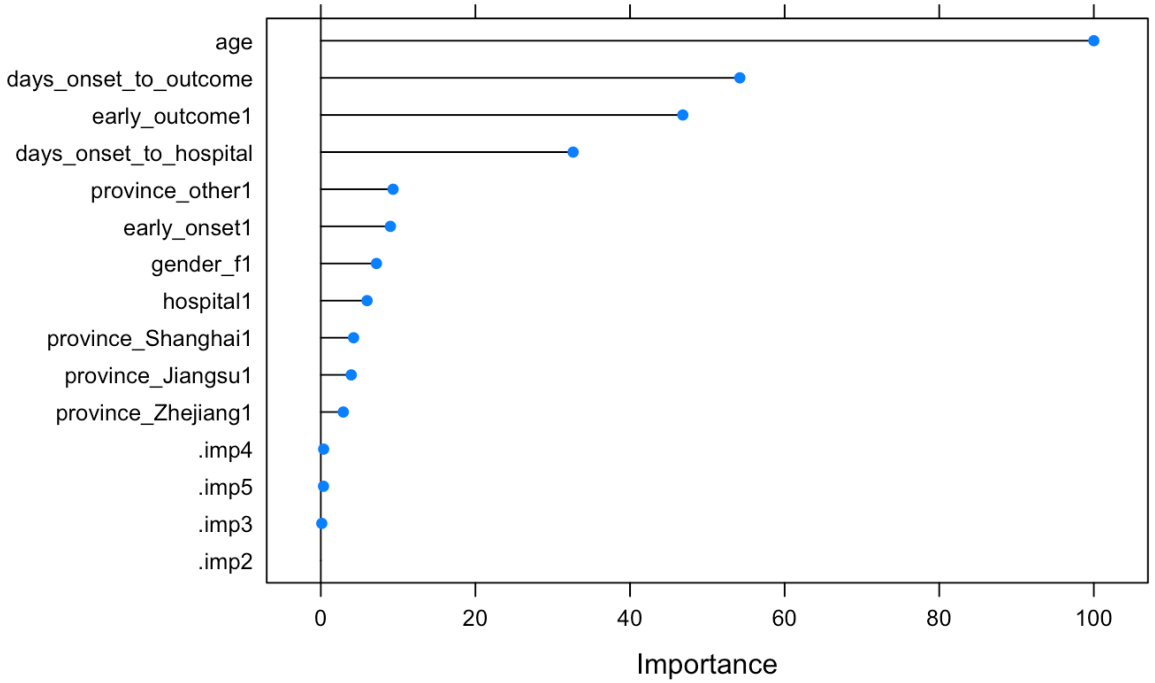


使用隨機森林建立模型

```
library(randomForest)
fit2 <- randomForest(outcome ~., data = val_train_data[,-1], ntree = 100)
predicted2 <- predict(fit2, val_test_data, type = 'class')
table(val_test_data$outcome, predicted2)
```

選出隨機森林模型特徵

```
control <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
set.seed(42)
model2 <- train(outcome ~ ., data = val_train_data[,-1], method = "rf", preProcess = NULL,
trControl = control)

importance2 <- varImp(model2, scale=TRUE)
plot(importance2)
```

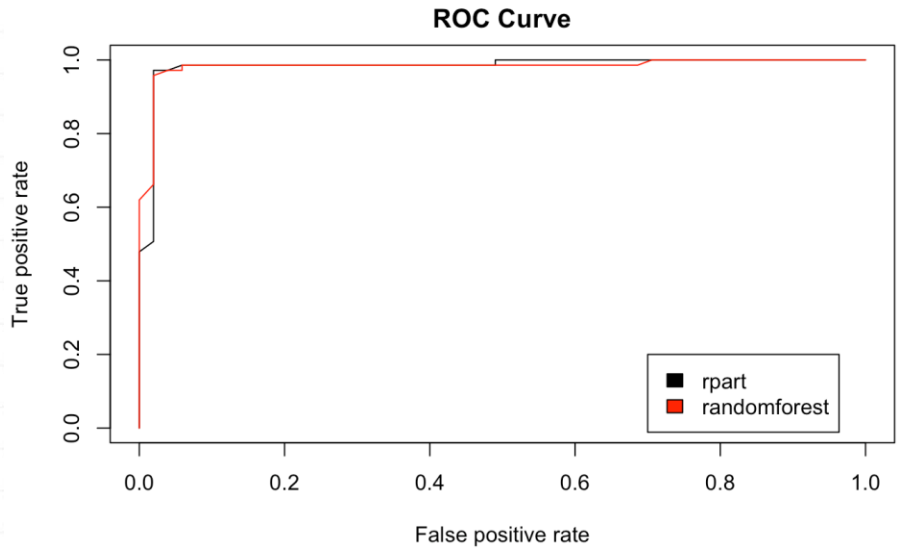


比較模型

```
# 決策樹
predictions1 <- predict(fit, val_test_data, type="prob")
pred.to.roc1 <- predictions1[, 2]
pred.rocr1 <- prediction(pred.to.roc1, as.factor(val_test_data$outcome))
perf.rocr1 <- performance(pred.rocr1, measure = "auc", x.measure = "cutoff")
perf.tpr.rocr1 <- performance(pred.rocr1, "tpr","fpr")

# 隨機森林
predictions2 <- predict(fit2, val_test_data, type="prob")
pred.to.roc2 <- predictions2[, 2]
pred.rocr2 <- prediction(pred.to.roc2, as.factor(val_test_data$outcome))
perf.rocr2 <- performance(pred.rocr2, measure = "auc", x.measure = "cutoff")
perf.tpr.rocr2 <- performance(pred.rocr2, "tpr","fpr")
```

比較ROC Curve

```
plot(perf.tpr.rocr1,main='ROC Curve', col=1)
legend(0.7, 0.2, c('rpart', 'randomforest'), 1:2)
plot(perf.tpr.rocr2, col=2, add=TRUE)
```



隨機森林表現比較好

參考資料

- A. Kucharski, H. Mills, A. Pinsent, C. Fraser, M. Van Kerkhove, C. A. Donnelly, and S. Riley. 2014. Distinguishing between reservoir exposure and human-to-human transmission for emerging pathogens using case onset data. PLOS Currents Outbreaks. Mar 7, edition 1. doi: 10.1371/currents.outbreaks.e1473d9bfc99d080ca242139a06c455f.

- A. Kucharski, H. Mills, A. Pinsent, C. Fraser, M. Van Kerkhove, C. A. Donnelly, and S. Riley. 2014. Data from: Distinguishing between reservoir exposure and human-to-human transmission for emerging pathogens using case onset data. Dryad Digital Repository. http://dx.doi.org/10.5061/dryad.2g43n.

THANK YOU