talk11 练习与作业

目录

0.1 练习和作业说明

将相关代码填写入以 “‘{r} “‘ 标志的代码框中，运行并看到正确的结果；

完成后，用工具栏里的”Knit” 按键生成 PDF 文档；

将 PDF 文档改为：姓名`-`学号`-talk11` 作业`.pdf`，并提交到老师指定的平台/钉群。

0.2 talk11 内容回顾

待写..

0.3 练习与作业：用户验证

请运行以下命令，验证你的用户名。

如你当前用户名不能体现你的真实姓名，请改为拼音后再运行本作业！

```
Sys.info()[["user"]]
```

```
## [1] "lucas"
```

```
Sys.getenv("HOME")
```

```
## [1] "/Users/lucas"
```

0.4 练习与作业 1：linear regression

0.4.1 一元回归分析

用 readr 包的函数将 Excercises and homework/data/talk11/ 目录下的 income.data_.zip 文件装入到 income.dat 变量中，进行以下分析：

1. 用线性回归分析 income 与 happiness 的关系；
2. 用点线图画出 income 与 happiness 的关系，将推导出来的公式写在图上；
3. 用得到的线性模型，以 income 为输入，预测 happiness 的值；
4. 用点线图画出预测值与真实 happiness 的关系，并在图上写出 R2 值。

```
## 代码写这里，并运行；
# Load the packages
library(readr)
library(ggplot2)

# Read the data
unzip("data/talk11/income.data_.zip", exdir = "data/talk11/")
income.dat =
  read_csv("data/talk11/income.data.csv",
           show_col_types = FALSE)
```

```
## New names:
## * `` -> `...1`
```

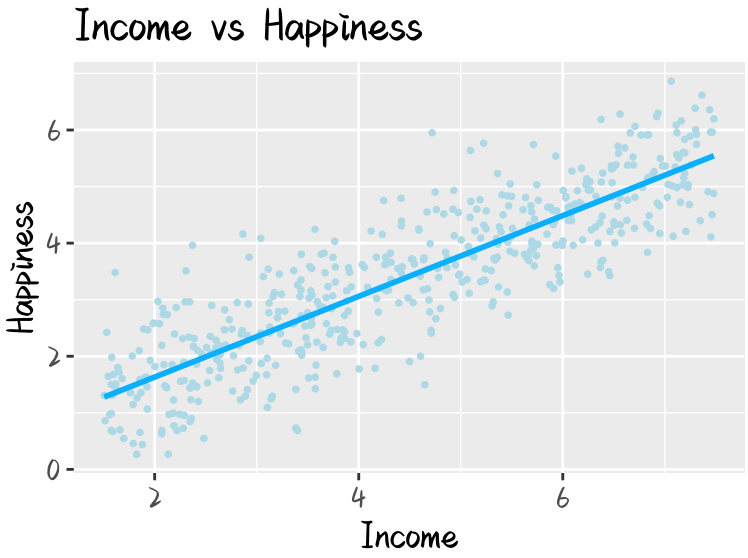
```
# Executing linear prediction
income_linear_model =
  lm(happiness ~ income,
     data = income.dat)
summary(income_linear_model)
```

```
##
## Call:
## lm(formula = happiness ~ income, data = income.dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.02479 -0.48526  0.04078  0.45898  2.37805
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.20427    0.08884   2.299   0.0219 *
## income       0.71383    0.01854  38.505   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7181 on 496 degrees of freedom
## Multiple R-squared:  0.7493, Adjusted R-squared:  0.7488
## F-statistic:  1483 on 1 and 496 DF,  p-value: < 2.2e-16
```

```
# Draw the plot
income_pic01 =
  ggplot(
    income.dat,
    aes(
      x = income,
```

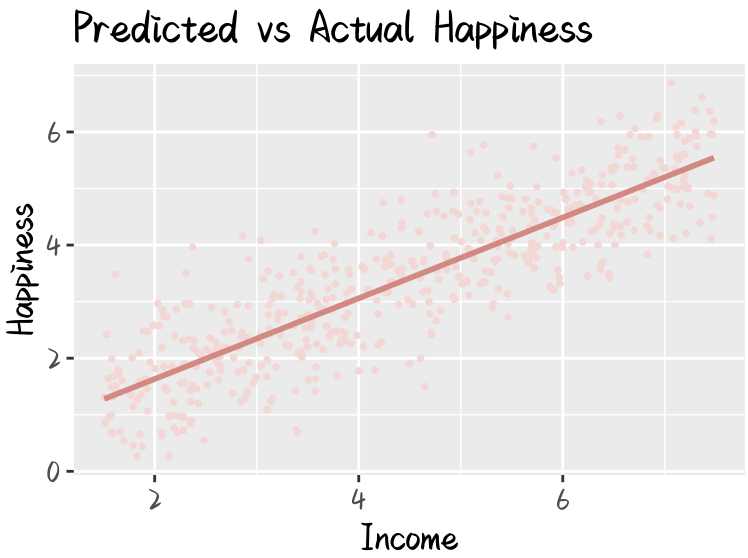
```
      y = happiness)) +
  geom_point(
    shape = 16,
    size = 1,
    color = "lightblue") +
  geom_smooth(
    method = "lm",
    se = FALSE,
    color = "#0BAFFF") +
  labs(
    title = "Income vs Happiness",
    x = "Income",
    y = "Happiness") +
  theme(
    text = element_text(
      family = "RLQDMSWR",
      size = 14))

print(income_pic01)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# Predict the value of happiness
income_predicted_happiness =
  predict(
    income_linear_model,
    newdata = data.frame(
      income = income.dat$income))

# Plotting predicted values against true values
income_df =
  data.frame(
    income = income.dat$income,
    happiness = income.dat$happiness,
    predicted = income_predicted_happiness)

income_pic02 =
  ggplot(income_df,
         aes(
           x = income,
           y = happiness)) +
```

```
    geom_point(
      shape = 16,
      size = 1,
      color = "#F2D7D4") +
    geom_line(
      aes(y = predicted),
      color = "#D38983",
      size = 1) +
    labs(
      title = "Predicted vs Actual Happiness",
      x = "Income",
      y = "Happiness") +
    theme(
      text = element_text(
        family = "RLQDMSWR",
        size = 14))

print(income_pic02)
```



0.4.2 多元回归分析

用 readr 包的函数将 Excercises and homework/data/talk11/ 目录下的 heart.data_.zip 文件装入到 heart.dat 变量中，进行以下分析：

1. 用线性回归分析 heart.disease 与 biking 和 smoking 的关系；
2. 写出三者间关系的线性公式；
3. 解释 biking 和 smoking 的影响（方向和程度）；
4. biking 和 smoking 能解释多少 heart.disease 的 variance？这个值从哪里获得？
5. 用 relaimpo 包的函数计算 biking 和 smoking 对 heart.disease 的重要性。哪个更重要？
6. 用得到的线性模型预测 heart.disease，用点线图画出预测值与真实值的关系，并在图上写出 R2 值。
7. 在建模时考虑 biking 和 smoking 的互作关系，会提高模型的 R2 值吗？如果是，意味着什么？如果不是，又意味着什么？

```
## 代码写这里，并运行；
# Load the packages
library(readr)
library(ggplot2)
library(relaimpo)

# Read the data
unzip("data/talk11/heart.data_.zip", exdir = "data/talk11/")
heart.dat =
  read_csv("data/talk11/heart.data.csv",
           show_col_types = FALSE)
```

```
## New names:
## * `` -> `...1`
```

```r
# Perform multiple linear regression
heart_linear_model =
  lm(heart.disease ~ biking + smoking,
     data = heart.dat)
summary(heart_linear_model)
```

```
##
## Call:
## lm(formula = heart.disease ~ biking + smoking, data = heart.dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1789 -0.4463  0.0362  0.4422  1.9331
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.984658   0.080137  186.99   <2e-16 ***
## biking      -0.200133   0.001366 -146.53   <2e-16 ***
## smoking      0.178334   0.003539   50.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.654 on 495 degrees of freedom
## Multiple R-squared:  0.9796, Adjusted R-squared:  0.9795
## F-statistic: 1.19e+04 on 2 and 495 DF,  p-value: < 2.2e-16
```

```r
# Calculate R2
R2 =
  summary(heart_linear_model)$r.squared
cat("R-squared value:", R2, "\n")
```

```
## R-squared value: 0.9796175
```

```r
# Calculating the importance of a variable
heart_data_importance =
  calc.relimp(heart_linear_model)
print(heart_data_importance)
```

```
## Response variable: heart.disease
## Total response variance: 20.90203
## Analysis based on 498 observations
##
## 2 Regressors:
## biking smoking
## Proportion of variance explained by model: 97.96%
## Metrics are not normalized (rela=FALSE).
##
## Relative importance metrics:
##
##               lmg
## biking  0.8795662
## smoking 0.1000512
##
## Average coefficients for different model sizes:
##
##                 1X        2Xs
## biking  -0.1990914 -0.2001331
## smoking  0.1704843  0.1783339
```

```r
# Performing multiple linear regressions
# including interactions
heart_interaction_model =
  lm(heart.disease ~ biking * smoking,
     data = heart.dat)

# Print summary of regression
```

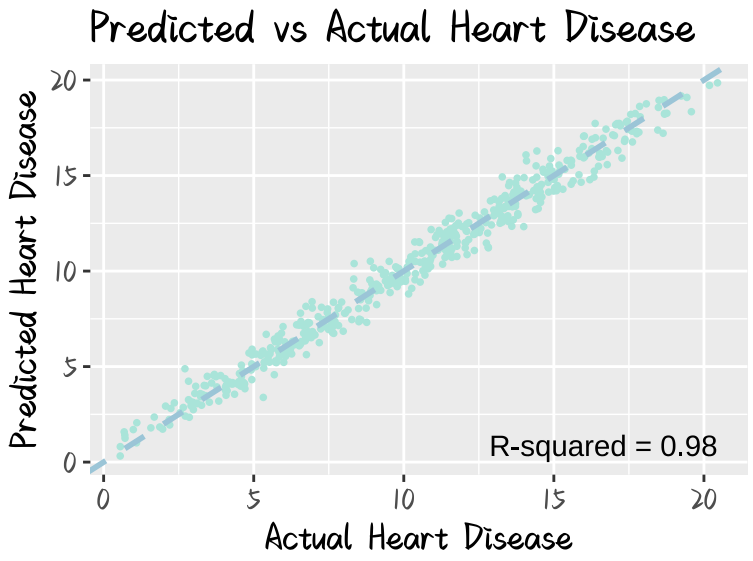
```r
# results with interactions
summary(heart_interaction_model)
```

```
##
## Call:
## lm(formula = heart.disease ~ biking * smoking, data = heart.dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.20619 -0.44862  0.02892  0.44099  1.94142
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    15.0527397  0.1248112 120.604   <2e-16 ***
## biking         -0.2019916  0.0029472 -68.536   <2e-16 ***
## smoking         0.1740065  0.0070359  24.731   <2e-16 ***
## biking:smoking  0.0001177  0.0001653   0.712    0.477
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6544 on 494 degrees of freedom
## Multiple R-squared:  0.9796, Adjusted R-squared:  0.9795
## F-statistic:  7922 on 3 and 494 DF,  p-value: < 2.2e-16
```

```r
# Predicting the value of heart.disease
predicted_heart.disease =
  predict(heart_linear_model,
          newdata =
            data.frame(
              biking = heart.dat$biking,
              smoking = heart.dat$smoking))

# Plotting predicted values against true values
```

```r
heart_df =
  data.frame(
    heart.disease = heart.dat$heart.disease,
    predicted = predicted_heart.disease)
heart_pic01 =
  ggplot(
    heart_df,
    aes(
      x = heart.disease,
      y = predicted)) +
    geom_point(
      shape = 16,
      size = 1,
      color = "#A9E4D9") +
    geom_abline(
      intercept = 0,
      slope = 1,
      linetype = "dashed",
      color = "#9BC5D7",
      size = 1) +
    labs(
      title = "Predicted vs Actual Heart Disease",
      x = "Actual Heart Disease",
      y = "Predicted Heart Disease") +
    theme(
      text = element_text(
        family = "RLQDMSWR",
        size = 14)) +
    annotate(
      "text",
      x = max(heart_df$heart.disease),
      y = min(heart_df$predicted),
      label =
```

```
        paste(
          "R-squared =",
          round(R2, 3)),
      hjust = 1,
      vjust = 0)

print(heart_pic01)
```



0.4.3 glm 相关问题

用 glm 建模时使用 family=binomial；在预测时，type= 参数可取值 link（默认）和 response。请问，两者的区别是什么？请**写代码**举例说明。

在 glm 中，type 参数在预测时用于选择输出的类型。具体而言，对于二项分布（family=binomial），type 参数可以设置为 link 或 response。

1. link：这是默认选项。返回的是线性预测的值，即链接函数（logit）的输出。在二项分布的情况下，这通常是 log-odds 的值。

2. response：返回的是估计的概率，即反映了响应变量为 1 的概率。

以下是使用 glm 建模和预测的示例代码：

```
## 代码写这里，并运行；

# Suppose there is dichotomous data,
# represented by the response variable

# In this example,
# family=binomial is used to
# indicate that the response variable is
# binomially distributed

# Generate sample data
set.seed(123)
example_data =
  data.frame(
    response =
      sample(c(0, 1),
             100,
             replace = TRUE),
    predictor1 = rnorm(100),
    predictor2 = rnorm(100)
  )

# Model
example_model =
  glm(
    response ~ predictor1 + predictor2,
    family = binomial,
    data = example_data)

# Create new data for prediction
```

```
example_new_data =
  data.frame(
    predictor1 = rnorm(10),
    predictor2 = rnorm(10)
  )

# Predicts and outputs LINK
example_link_predictions =
  predict(example_model,
          newdata = example_new_data,
          type = "link")
cat(
  "Link Predictions:",
  example_link_predictions,
  "\n"
)
```

```
## Link Predictions: -0.1575072 -0.1611032 -0.2873127 -0.5804015 -0.834559 -0.6332762 -
```

```
# Predict and output response
example_response_predictions =
  predict(example_model,
          newdata = example_new_data,
          type = "response")
cat(
  "Response Predictions:",
  example_response_predictions,
  "\n")
```

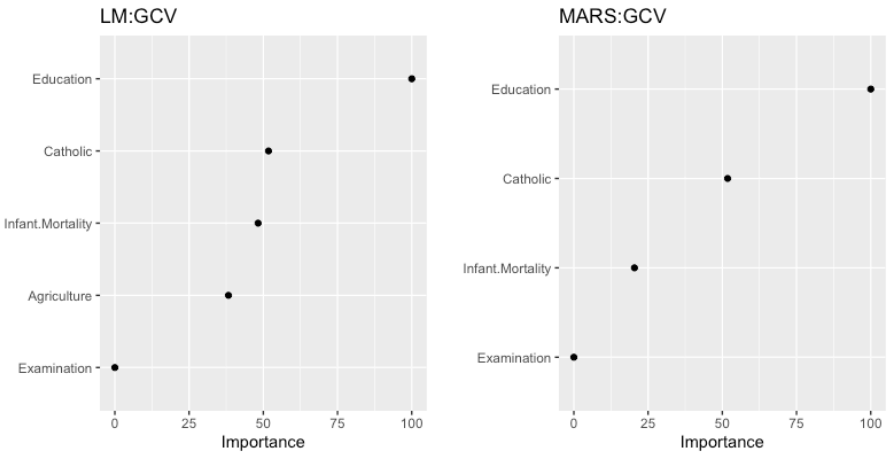
```
## Response Predictions: 0.4607044 0.4598111 0.4286619 0.3588402 0.302682 0.346768 0.43
```

在这个例子中，link_predictions 将包含线性预测的值，而 response_predictions 将包含估计的概率。其中，response_predictions 的值将在 0 到 1 之间，表示相应的二项分布中响应变量为 1 的概率。

0.5 练习与作业 2：non-linear regression

0.5.1 分析 swiss ，用其它列的数据预测 Fertility

1. 使用 earth 包建模，并做 10 times 10-fold cross validation;
2. 使用 lm 方法建模，同样做 10 times 10-fold cross validation;
3. 用 RMSE 和 R2 两个指标比较两种方法，挑选出较好一个;
4. 用 vip 包的函数查看两种方法中 feature 的重要性，并画图（如下图所示）:



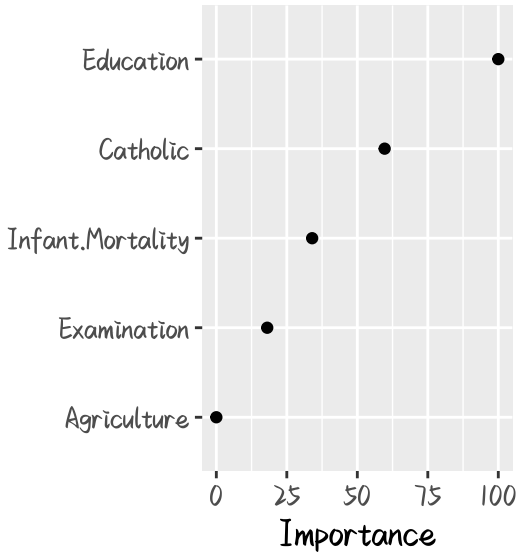

```
## 代码写这里，并运行;
# Loading the library
library(earth)
library(caret)
library(vip)

# Load the data
data(swiss)

# Setting control parameters
```

```r
# for cross-validation
ctrl =
  trainControl(
    method = "repeatedcv",
    number = 10,
    repeats = 10)

# Modeling using the EARTH method
model_earth =
  train(Fertility ~ .,
        data = swiss,
        method = "earth",
        metric = "RMSE",
        trControl = ctrl)

# Modeling using the LM approach
model_lm =
  train(Fertility ~ .,
        data = swiss,
        method = "lm",
        metric = "RMSE",
        trControl = ctrl)

# Comparing the results of the two models
resamples(
  list(Earth = model_earth,
       LM = model_lm))
```

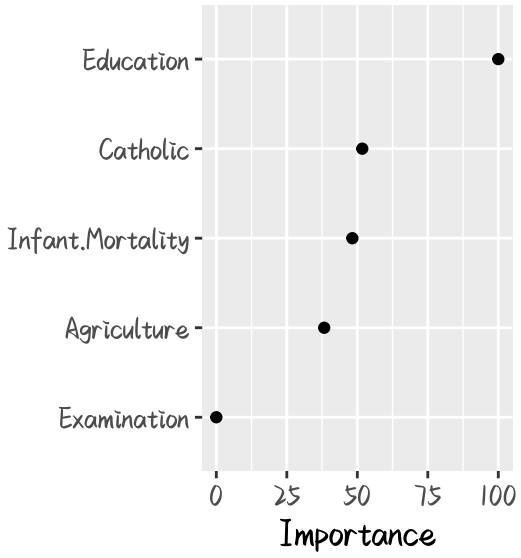
```
##
## Call:
## resamples.default(x = list(Earth = model_earth, LM = model_lm))
##
## Models: Earth, LM
```

```
## Number of resamples: 100
## Performance metrics: MAE, RMSE, Rsquared
## Time estimates for: everything, final model fit
```

```
# Mapping the importance of
# features for the EARTH model
earth_model_plot =
  vip(model_earth,
      geom = "point")

earth_model_pic01 =
  earth_model_plot +
    geom_point(
    shape = 16,
    size = 1) +
    theme(
    text = element_text(
      family = "RLQDMSWR",
      size = 14))

print(earth_model_pic01)
```




```
# Mapping the importance of
# features for the LM model
lm_model_plot =
  vip(model_lm,
      geom = "point")

lm_model_pic01 =
  lm_model_plot +
    geom_point(
    shape = 16,
    size = 1) +
    theme(
    text = element_text(
      family = "RLQDMSWR",
      size = 14))

print(lm_model_pic01)
```


Education
Catholic
Infant.Mortality
Agriculture
Examination
0
25
50
75
100
Importance