2019_final_exam_prediction

刘栋梁

2019年6月20日

Preface

对2019年生统考试数据的预测，主要总结群里大神们的讨论和代码。

注意写原假设、备择假设以及结论。

有错误的地方直接在群里立即指出。

Question 2

2.在一个抗焦虑症药理学实验中，研究人员测试了三种不同的药物和安慰剂对小鼠行为的影响，统计小鼠啃咬垫料、移动频率多种行为学数据并打分，分数越高则抗焦虑效果越好，结果记录在q2_data.csv中。

请问：

（1）药物的种类是否影响小鼠的表现；

用方差分析方法检验药物的种类是否影响小鼠的表现。

H_0：服用药物的种类不会影响小鼠的表现；

H_1：至少一种药物影响小鼠的表现。

```
drug <- read.csv("./data/q2_data.csv")

shapiro.test(subset(drug, drug$Treatment=="excitingDrug")$Performance_Scores)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(drug, drug$Treatment == "excitingDrug")$Performance_Scores
## W = 0.98952, p-value = 0.978
```

```
shapiro.test(subset(drug, drug$Treatment=="anxietyDrug")$Performance_Scores)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(drug, drug$Treatment == "anxietyDrug")$Performance_Scores
## W = 0.97895, p-value = 0.9289
```

```
shapiro.test(subset(drug, drug$Treatment=="hypertensionDrug")$Performance_Scores)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(drug, drug$Treatment == "hypertensionDrug")$Performance_Scores
## W = 0.97348, p-value = 0.8971
```

```
shapiro.test(subset(drug, drug$Treatment=="Placebos")$Performance_Scores)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  subset(drug, drug$Treatment == "Placebos")$Performance_Scores
## W = 0.98952, p-value = 0.978
```

```
bartlett.test(Performance_Scores~Treatment, data = drug)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  Performance_Scores by Treatment
## Bartlett's K-squared = 0.13278, df = 3, p-value = 0.9876
```

```
ff <- aov(Performance_Scores~Treatment, data = drug)
anova(ff)
```

```
## Analysis of Variance Table
##
## Response: Performance_Scores
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## Treatment  3 14353.0  4784.3  39.516 1.263e-07 ***
## Residuals 16  1937.2   121.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

答：检验结果显示p值小于0.05，因此拒绝原假设，药物的种类会影响小鼠的表现。

这个有可能让你用手算方差分析表。

手算方差分析表：

Source of variation	SS	*df*	MS	*F* statistic	*p*-value
Between	$\sum_{j=1}^{k} n_j \bar{X}_j^2 - \frac{\bar{\bar{X}}^2}{n} = A$	$k-1$	$\frac{A}{k-1}$	$\frac{A/(k-1)}{B/(n-k)} = F$	$Pr(F_{k-1,n-k} > F)$
Within	$\sum_{j=1}^{k} (n_j - 1) {s_j}^2 = B$	$n-k$	$\frac{B}{n-k}$		
Total	Between SS + Within SS				

$$Between\ SS = \sum_{j=1}^{k} n_j \bar{X}_j^2 - \frac{\bar{\bar{X}}^2}{n}$$

$$Within\ SS = \sum_{j=1}^{k} (n_j - 1) {s_j}^2$$

```
#import data
anova_data<- read.csv("./data/q2_data.csv",sep=",")
anova_data$Treatment<-factor(anova_data$Treatment)
attach(anova_data)
GrandMean<-mean(Performance_Scores);GrandMean
```

```
## [1] 57.3
```

```
SMeans<-aggregate(Performance_Scores,by=list(Treatment),FUN=mean);SMeans
```

```
##            Group.1    x
## 1         Placebos 16.6
## 2      anxietyDrug 50.0
## 3     excitingDrug 83.4
## 4 hypertensionDrug 79.2
```

```
# (2) Compute Sum of Squares
SVars<-aggregate(Performance_Scores,by=list(Treatment),FUN=var);SVars
```

```
##            Group.1     x
## 1         Placebos 112.3
## 2      anxietyDrug 109.0
## 3     excitingDrug 112.3
## 4 hypertensionDrug 150.7
```

```
SLens<-aggregate(Performance_Scores,by=list(Treatment),FUN=length);SLens
```

```
##            Group.1 x
## 1         Placebos 5
## 2      anxietyDrug 5
## 3     excitingDrug 5
## 4 hypertensionDrug 5
```

```
within_SS <- sum((SLens$x-1)*SVars$x);within_SS
```

```
## [1] 1937.2
```

```
total_SS <- sum((Performance_Scores-GrandMean)^2);total_SS
```

```
## [1] 16290.2
```

```
between_SS <- total_SS-within_SS;between_SS
```

```
## [1] 14353
```

```
# (3)Compute Degree of Freedom
df_between <- length(levels(Treatment))-1;df_between
```

```
## [1] 3
```

```
df_within <-length(Performance_Scores)-length(levels(Treatment));df_within
```

```
## [1] 16
```

```
# (4)Compute Mean Square
between_MS <- between_SS/df_between;between_MS
```

```
## [1] 4784.333
```

```
within_MS <- within_SS/df_within;within_MS
```

```
## [1] 121.075
```

```
# (5)F-ration and p-value
F_ratio <- between_MS/within_MS;F_ratio
```

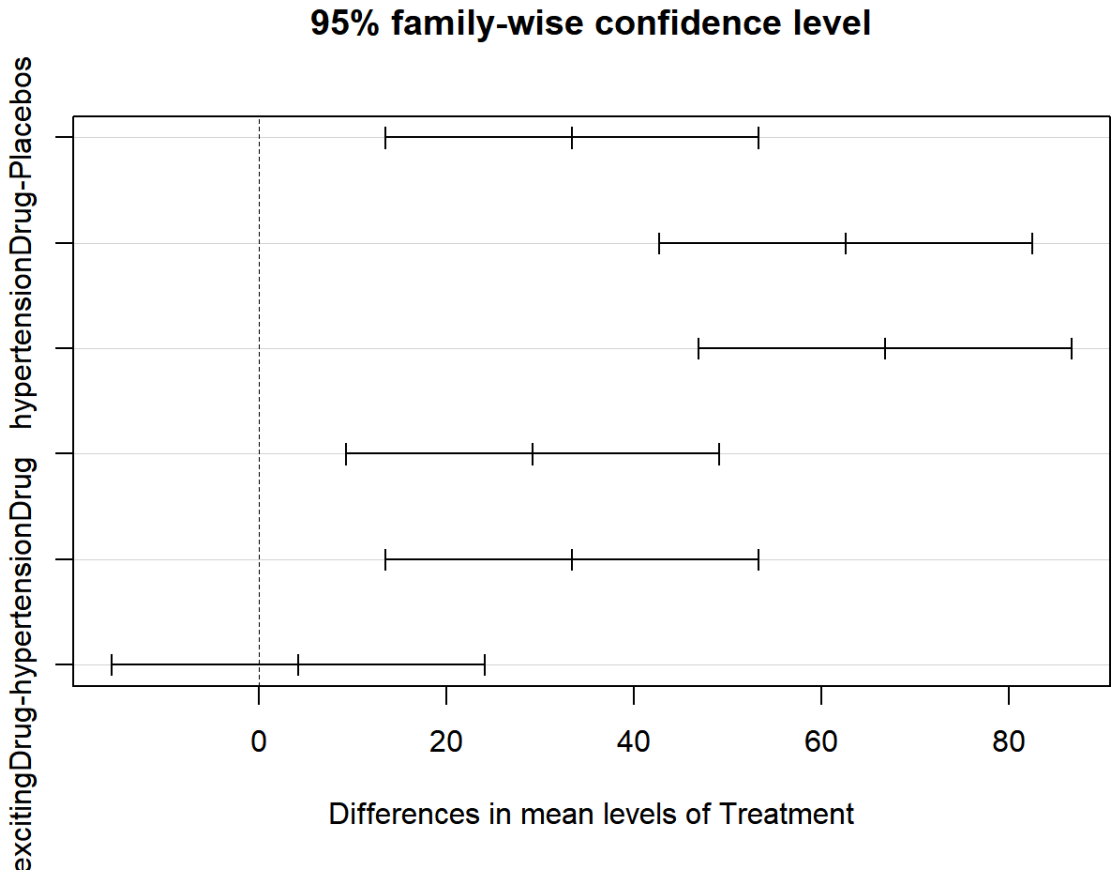
```
## [1] 39.51545
```

```
p_value <-1-pf(F_ratio,df_between,df_within);p_value
```

```
## [1] 1.262592e-07
```

(2).如果受药物影响，那么哪一个药物效果最好。

```
# 多重比较
tukey <- TukeyHSD(ff,ordered = TRUE)
plot(tukey)
```



答：hypertensionDrug-excitingDrug之间是不显著差异，其他都是显著的。(包含零为不显著)

Question 3

> 这个相似度是最高的，不过不能确认问题是不是一样。

Many digitized image of a fine needle aspirate (FNA) of a breast mass are collected and computed to predict the diagnosis of breast cancer(q3_BC.txt).

Attribute information

1. Diagnosis (1 = malignant, 0 = benign)

Ten real-valued features are computed for each cell nucleus:

```
a) radius (mean of distances from center to points on the perimeter)

b) texture (standard deviation of gray-scale values)

c) perimeter

d) area

e) smoothness (local variation in radius lengths)

f) compactness (perimeter^2 / area - 1.0)

g) concavity (severity of concave portions of the contour)

h) concave points (number of concave portions of the contour)

i) symmetry

j) fractal dimension ("coastline approximation" - 1)
```

The mean of these features were computed for each image.

> 这里和作业不一样，简化了，只取了均值。

All feature values are recorded with four significant digits. In total, there are 357 benign and 212 malignant samples.

You may need to use proper regression algorithm to train your data, and make predictions.

Instructions:

1)Use all mean features(such as: radius_mean,texture_mean…) to construct a logistic regression model.

```
q3_BC <- read.table('./data/q3_BC.txt',header = T)
fit.full <- glm(diagnosis~.,data = q3_BC,family =  binomial()) # ~.即代表所有模型
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(fit.full)
```

```
##
## Call:
## glm(formula = diagnosis ~ ., family = binomial(), data = q3_BC)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.95590  -0.14839  -0.03943   0.00429   2.91690
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -7.35952   12.85259  -0.573   0.5669
## radius             -2.04930    3.71588  -0.551   0.5813
## texture             0.38473    0.06454   5.961  2.5e-09 ***
## perimeter          -0.07151    0.50516  -0.142   0.8874
## area                0.03980    0.01674   2.377   0.0174 *
## smoothness         76.43227   31.95492   2.392   0.0168 *
## compactness        -1.46242   20.34249  -0.072   0.9427
## concavity           8.46870    8.12003   1.043   0.2970
## concave.points     66.82176   28.52910   2.342   0.0192 *
## symmetry           16.27824   10.63059   1.531   0.1257
## fractal_dimension -68.33703   85.55666  -0.799   0.4244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 146.13  on 558  degrees of freedom
## AIC: 168.13
##
## Number of Fisher Scoring iterations: 9
```

> glm.fit:拟合機率算出来是数值零或一:可以直接忽略这个警告。这个问题实际是因为数据集问题，这个数据集本身接近线性可分了，所以导致模型过拟合。具体原因大家自己检索一下吧。可以看看这篇博客：https://www.cnblogs.com/runner-ljt/p/4574275.html (https://www.cnblogs.com/runner-ljt/p/4574275.html)。

2)Then try to reduce the number of features from your last model, construct another regression model, and you will need to write down the equation of your logistic regression model(Tips:

$$LogitP = \alpha + \beta_1 X_1 + \beta_2 X_2 + . . + \beta_p X_p$$

)

除去不显著相关的变量。

```
fit.reduced<-glm(diagnosis~texture+area+smoothness+concave.points,data = q3_BC,family = binomial())
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(fit.reduced)
```

```
##
## Call:
## glm(formula = diagnosis ~ texture + area + smoothness + concave.points,
##     family = binomial(), data = q3_BC)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.31798  -0.15623  -0.04212   0.01662   2.84201
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -23.677816   3.882774  -6.098 1.07e-09 ***
## texture          0.362687   0.060544   5.990 2.09e-09 ***
## area             0.010342   0.002002   5.165 2.40e-07 ***
## smoothness      59.471304  25.965153   2.290    0.022 *
## concave.points  76.571210  16.427864   4.661 3.15e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.44  on 568  degrees of freedom
## Residual deviance: 156.44  on 564  degrees of freedom
## AIC: 166.44
##
## Number of Fisher Scoring iterations: 8
```

公式:

$$LogitP = -23.677816 + 0.362687x_{texture} + 0.010342x_{area} + 59.47130x_{smoothness} + 76.571210x_{concave.points}$$

> 注意和odds以及p的公式形式的区别。

3)Use proper test to test the difference between two models

对两个模型进行ANOVA分析。

```
#H0:两模型无显著差别；HA:两模型有显著差别
anova(fit.full,fit.reduced,test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: diagnosis ~ radius + texture + perimeter + area + smoothness +
##     compactness + concavity + concave.points + symmetry + fractal_dimension
## Model 2: diagnosis ~ texture + area + smoothness + concave.points
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       558     146.13
## 2       564     156.44 -6   -10.31   0.1122
```

p=0.1122>0.05，两模型无显著差别。

4)You may split the data properly, use part of them to train your regression model and use another part to make predictions. Lastly, you may try to calculate the accuracy of your model.(Tips: To split the data, you can use the first 398 rows as training data, use the last 171 rows as prediction data.The predict function return a value between 0 and 1, 0.~0.5 belong to the first class, and 0.5~1 belong to second class in binary classification problems)

```
train_data <- q3_BC[1:398,]
test_data <- q3_BC[399:569,]
train_fit<-glm(diagnosis~texture+area+smoothness+concave.points,data = train_data,family = binomial())
summary(train_fit)
```

```
##
## Call:
## glm(formula = diagnosis ~ texture + area + smoothness + concave.points,
##     family = binomial(), data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.39278  -0.14454  -0.02447   0.03635   2.60665
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -27.47397    4.74798  -5.786 7.19e-09 ***
## texture          0.46244    0.08434   5.483 4.19e-08 ***
## area             0.01082    0.00235   4.606 4.11e-06 ***
## smoothness      90.11221   30.96961   2.910 0.003618 **
## concave.points  59.01212   17.51779   3.369 0.000755 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 544.93  on 397  degrees of freedom
## Residual deviance: 108.30  on 393  degrees of freedom
## AIC: 118.3
##
## Number of Fisher Scoring iterations: 8
```

```
test_data$probe<-predict(train_fit,newdata = test_data,type = "response")
lables <- test_data$diagnosis
pred_lables <- ifelse(test_data$probe > 0.5, 1, 0)
mean(pred_lables==lables)
```

```
## [1] 0.9064327
```

The accuracy is 0.9064

Question 4

Alzheimer's Disease (AD) is the most common cause of dementia, a group of brain disorders that results in the loss of intellectual and social skills. These changes are severe enough to interfere with day-to-day life. AD is characterized by the presence of senile plaques and neurofibrillary tangles in cortical regions of the brain. These pathological markers are thought to be responsible for the massive cortical neurodegeneration and concomitant loss of memory, reasoning, and after aberrant behaviors that are seen in patients with AD.

Here we have a gene expression data from normal neurons and neurons containing neurofibrillary tangles of 14 mid-stage AD cases. It can be found in "expression_data.txt". Column 1-7 of "expression_data.txt" are normal neurons and column 8-14 are neurons with neurofibrillary tangles. Use the information mentioned above to answer the following questions:

a)Use t-test to find significantly differential expression genes between normal and tangle neuron sample (p-value < 0.01). Give the number of differential expressed genes and give the names of significantly differential expression genes.

Hint1: two types of t-test with equal variance and unequal variance are different.

Hint2: “names()” or “rownames()” can be used to extract names of differentially expressed genes.

Hint3: “apply(data,1,function(x){…})” can apply function to every row in data more quickly than “for{}”, so try to use “apply”.

```
q4_data <- read.table("./data/q4_data.txt", header = T,stringsAsFactors = T)
#check NA
table(is.na(q4_data))
```

```
##
## FALSE
##  3806
```

```
# 个人觉得应该先检查NA，毕竟标签缺失了应该算缺失值。
exp_matrix <- t(q4_data[2:11]) # t()函数即转置
colnames(exp_matrix) <- q4_data$Group

# 邮件没有提示用limma包，可能会让我们自己处理标准化。具体是什么标准化方法大家按题目要求做吧。这里简单地除以均值。TIN:总强度标准化方法。

exp_matrix_norm<-apply(exp_matrix,2, function(x){x/mean(x)})

# 计算方差齐性检验的p值
p.var.test<-apply(exp_matrix_norm,1,function(x) var.test(x[1:161],x[162:346],conf.level = 0.99)$p.value)

# 把p值结果加到数据后面
exp_matrix_norm_p<- cbind(exp_matrix_norm,p.var.test)

# 注意这边less、greater还是双尾检测需要根据题目来。

# p.t.value <-  apply(exp_matrix_norm_p,1, function(x) t.test(x[1:161],x[162:346],alternative=c("less"),var.equal  =  x[347]>=0.01)$p.value)

p.t.value  <-  apply(exp_matrix_norm_p,1, function(x) t.test(x[1:161],x[162:346],var.equal =  x[347]>=0.01)$p.value)

sum(p.t.value<0.01)
```

```
## [1] 10
```

```
sort(p.t.value[p.t.value<0.01])
```

```
##        AP5B1        ADAM8         APIP     ADAMTSL4         AFF3
## 0.0008445342 0.0009474259 0.0012923281 0.0014098548 0.0016624990
##        AKAP5        ACTN4         ARF4          ABR      ARHGAP4
## 0.0022132731 0.0030026255 0.0036146269 0.0044860337 0.0047948672
```

b)Adjust the p-values in question a) with both “bonferroni” and “fdr” method to find differentially expressed genes (adjusted p-value < 0.01). Give the number of differential expressed genes.

Hint: you can do the adjustment according to the formula, or use “p.adjust()” instead

```
p.bonf <- p.adjust(p.t.value,'bonferroni')
sum(p.bonf<0.01)
```

```
## [1] 2
```

```
sort(p.bonf[p.bonf<0.01])
```

```
##       AP5B1       ADAM8
## 0.008445342 0.009474259
```

```
p.fdr <- p.adjust(p.t.value,'fdr')
sum(p.fdr<0.01)
```

```
## [1] 10
```

```
sort(p.fdr[p.fdr<0.01])
```

```
##       ADAM8    ADAMTSL4        AFF3       AP5B1        APIP       AKAP5
## 0.003324998 0.003324998 0.003324998 0.003324998 0.003324998 0.003688789
##       ACTN4        ARF4         ABR     ARHGAP4
## 0.004289465 0.004518284 0.004794867 0.004794867
```