Test your knowledge of logistic regression in R

1) baseballlarge ← read.csv("baseballlarge.csv")

a) str(baseballlarge)

i) There are a total of 1232 observations (team/year pairs) in the dataset.

ii) table(baseballlarge$Year)

length(table(baseballlarge$Year))

There are a total of 47 years in the dataset though it ranges from 1962 to 2012 (missing are 1972, 1981, 1994, 1995)

iii) baseballlarge ← subset(baseballlarge, Playoffs == 1)

Str(baseballlarge)

There are a total of 244 team/year pairs in the new data frame.

iv) table(baseballlarge$Year)

table(table(baseballlarge$Year))

2	4	8	10	} No of teams in playoffs
7	23	16	1	} No of years

b)

baseballlarge$NumCompetitors <-

table(baseballlarge$Years)[as.character(baseballlarge$Year)]

Note that to obtain a particulare entry from the table, we need to call it with table(baseballlarge$Year)["1962"] or [as.character(1962

table(baseballlarge$NumCompetitors)

This gives 128 playoff team/year pairs where 8 teams were invited to playoffs. Note from a)iv) this can also be verified as $8(16) = 128$.

c) baseballlarge$WorldSeries <-

as.integer(baseballlarge$RankPlayoffs == 1)

table(baseballlarge$WorldSeries)

0	1
197	47

There are 197 teams in the years dataset where teams/year pair did not win the World Series

d)

```
Model 1 <- glm ( WorldSeries ~ Year, data = baseballlarge,
                 family = binomial )

Model 2 <- glm ( WorldSeries ~ RS, data = baseballlarge,
                 family = binomial )

   :    ( Do similarly for RA, W, OBP,
          SLG, BA, RankSeason,
          NumCompetitors )
```

For the League variable we do,

```
baseballarge$League <- as.integer ( baseballarge$League
                                    == "AL" )

Model 10 <- glm ( WorldSeries ~ League,
                  data = baseballlarge, family = binomial )
```

Note that here AL is stored as 1 & NL as 0.

Since there are only 2 categories & unordered, this approach makes sense of handling the independent variable..

```
Summary (model 1)
   :
Summary (model 10)
```

From the results, we find the following variables are significant:

1) Year,
2) RA,
3) RankSeason,
4) Num Competitors

W just misses out with p-value = 0.0577
SLG also p-value = 0.0504

e) Note while we will include Year, RA, RankSeason & Num Competitors, from a modeling perspective, it also makes sense to include W & SLG since they have p-values close to 0.05.

```
modelsig <- glm(WorldSeries ~ Year + RA +
                RankSeason + NumCompetitors, data =
                baseballlarge, family = binomial)

Summary(modelsig)
```

In the new multivariate model, none of the variables are significant.

f) `cor(baseballlarge[, c("Year", "RA", "RankSeason", "NumCompetitors")])`

In this example, the year and number of Competitors has a high correlation

g)

```
model g1 <- glm (World Series ~ Year + RA,
                 data = baseballlarge, family = binomial)

model g2 <- glm (World Series ~ Year + Rank season,
                 data = baseballlarge, family = binomial)

model g3 <- glm (World Series ~ Year +
                 Num Competitors, data = baseballlarge,
                 family = binomial)

model g4 <- glm (World Series ~ RA + Rank season,
                 data = baseballlarge, family = binomial)

model g5 <- glm (World Series ~ RA + Num Competitors,
                 data = baseballlarge, family = binomial)

model g6 <- glm (World Series ~ Rank season +
                 Num Competitors, data = baseballlarge,
                 family = binomial)

Summary (model g1)
    :
Summary (model g6)
```

The smallest AIC value corresponds to model 9

World Seris ~ Num Competitors.

h) Overall it seems the winning of playoffs has a large role of luck & other season performances do not

2)

a) Parole ← read.csv("Parole.csv")

Str(Parole)

There are a total of 675 parolees in this date.

b) table(Parole $ Violator)

0	1
597	78

78 of the parolees violated their terms of parole.

c) In this date, State & Crime are unordered factor variables with at least 3 variables

d) set.seed(144)

library(caTools)

Split ← Sample.split(Parole $ Violator, SplitRatio = 0.7)

train ← subset(Parole, Split == TRUE)

test ← subset(Parole, Split == FALSE)

In the split, roughly 70% of the parolees have been assigned to the training & 30% to the test set.

If you rerun the commands (1) - (5), we would expect to get the same training / test split as the first execution. This is because we set the seed (random number generator) to be the same.

If we just rerun (3) - (5), we would get a different training / test split.

If we set seed to a different number from 144 & rerun (3) - (5), we would expect to see a different training / test split.

(R) model1 <- glm(Violator ~ ., data = train, family = binomial)

Summary(model1)

The significant variables at the 0.05 level are: RaceWhite (race of parolee), State Virginia, Multiple offenses.

f) From the logistic regression result, we get:

β for multiple offense = 1.61.

Everything else being equal, we have

$$\text{Odds} = \frac{P(Y=1)}{P(Y=0)} = e^{\beta \times (\text{Multiple offenses})}$$

If a person did not commit multiple offenses this number Multiple offenses = 0, else 1.

$$\therefore \text{Odds} = e^{\beta(1)} = e^{1.61} = 5.01$$

Of parolee to be a violator with multiple offenses compared to a person who did not commit

Thus Statement 4) is the appropriate one.

g) Odds of the given individual being a violator

$$= e^{-2.03 + 0.38(1) + 0.88(1) - 0.0017(50) - 0.12(3) + 0.08(12) + 1.66(0) + 0.695(1)}$$

(1) → Male; (1) → White; (50) → age; (3) → Time served; (12) → Max sentence; (0) → Multiple offense; (1) → larceny

We can evaluate these numbers as follows:

We can compute the odds & log odds as.

logodds ← model1$coef[1] + model1$coef[2] × 1 (Male)
(model1$coef[1]: Intercept)

+ model1$coef[3] × 1 (White) + model1$coef[4] × 50 (Age)

+ model1$coef[5] × 3 (Time served) + model1$coef[9] × 12 (Max Sentence)

+ model1$coef[10] × 0 (Multiple Offense) + model1$coef[12] × 1 (Committed larceny)

Odds ← exp(logodds)

Odds of the individual being a violator = 0.28

& log odds = −1.25

h)

```
pred <- predict(model1, newdata = test,
                type = "response")

max(pred)
```

Maximum predicted probability of violation = 0.907

i)

```
table(as.integer(pred > 0.5), test$Violator)
```

Predicted \ Actual	0	1
0	167	11
1	12	12

$$\text{Sensitivity} = \underbrace{TPR}_{\text{True positive rate}} = \frac{12}{12+11} = 0.521$$

$$\text{Specificity} = \underbrace{TNR}_{\text{True negative rate}} = \frac{167}{167+12} = 0.932$$

$$\text{Accuracy} = \frac{167+12}{167+12+12+11} = 0.886$$

j) table(test$Violator)

Model	0 (Actual)	1 (Actual)
0	179	23

If model predicts everyone as a non-violator

$$\text{Accuracy} = \frac{179}{179+23} = 0.886$$

Same as previous model

k) Clearly in this context false negatives are a worry where parolees who will be violators are released. Thus it is natural for the board to assign more cost to a false negative than a false positive and should use a cutoff less than 0.5.

Lowering the cutoff makes the model predict more people to be positive reducing the undesirable outcome thus

l) The model is of likely value to the board since it can provide a better characterization than the simple model. While both models have the same accuracy, the baseline model produces many false negatives (23 Compared to 11). Changing the threshold is likely to improve the model's value. Thus the last option is the most accurate assessment.

m)
```
library(ROCR)
predrocr <- prediction(pred, test$Violator)
auc <- performance(predrocr, measure = "auc")
@ y.values
```

The Area under Curve = 0.894

n) The AUC can be interpreted as the probability that the model can correctly differentiate between a randomly selected parole violator & a randomly selected parole non violator.

o) Option 2 does not capture the true outcome of parolees since they are still either in jail or not violated thus far. Option 3 dus not help us to build a better model. Option 4) is the best where they are tracked until they violate the parole or complete the term. However such a data set needs more effort to gather

3)

a)
```
germancredit <- read.csv("germancredit.csv")
set.seed(2016)
library(caTools)

spl <- sample.split(germancredit$resp, 0.75)

training <- subset(germancredit, spl==TRUE)

test <- subset(germancredit, spl==FALSE)
```

The reason for splitting the dataset in such a way is to ensure that the dependent variable is balanced between the training & test sets

This can be easily verified by:

```
table(training$resp)
```

0	1
225	525

$\frac{525}{525+225} = 0.7$

```
table(test$resp)
```

0	1
75	175

$\frac{175}{175+75} = 0.7$

b)
```
model1 <- glm(resp ~ 1, data = training,
              family = binomial)

Summary(model1)
```

This fits a logistic regression model only with the intercept. The fitted model is:

$$P(resp = 1) = \frac{e^{0.8473}}{1 + e^{0.8473}} = 0.7$$

c) Note that the result in part (b) is exactly equal to the fraction of the number of people in the traning set with a good credit rating.

d) model 2 ← glm(resp ~ . , data = traning ,
family = binomial)

Summary(model 2)

The vandles that are significant at the 10% level are:

Chckacct , dur , hist , amt , sav ,
instrate , malesingle , age , foreign (for.)

e) Residual deviance = $-2LL(\hat{\beta})$ = 689.55

$\therefore \underbrace{LL(\hat{\beta})}_{\text{Loglikelihood at estimated value}} = -344.775$

g) pred2 ← predict (model 2, newdata = test ,
type = "response")

table (pred2 > 0.5, test $resp)

Model \ Actual	0	1
0	40	18
1	35	157

Model accuracy $= \frac{157+40}{157+40+18+35} = 0.788$

h)

```
model3 <- glm(resp ~ chkacct + dur + hist +
              amt + sav + instrate + malesingle +
              age + for. - 1, data = training,
              family = binomial)

summary(model3)
```

Note that the foreign variable is stored as for. and -1 is used to remove the intercept.

AIC of model 3 = 750.24

i) AIC of model 2 = 751.55

In terms of AIC, model 3 is preferred to model 2.

j)

```
pred3 <- predict(model3, newdata = test,
                 type = "response")
table(pred3 >= 0.5, test$resp)
```

	Actual 0	Actual 1
0	33	15
1	42	160

Model

Note that the accuracy is lower in test set than model2

$$\text{Accuracy} = \frac{160+33}{160+33+42+15} = 0.772$$

k)
In model 2, the fraction of people who are predicted as good risk but are actually bad risks (Type I error) $= \frac{35}{35+40} = 0.4667$

In model 3, this fraction $= \frac{42}{42+33} = 0.56$

Clearly model 2 is preferred in this metric

l) In model 2, the fraction of people who are predicted as bad risk but are actually good risk (Type II error) $= \frac{18}{18+157} = 0.102$

In model 3, this fraction $= \frac{15}{15+160} = 0.0857$

Clearly model 3 is preferred in this metric.

m) library (ROCR)
predrocr2 ← Prediction (pred2, test$resp)
auc2 ← performance (predrocr2, measure = "auc") @ y.values

predrocr3 ← prediction (pred3, test$resp)
auc3 ← performance (predrocr3, measure = "auc") @ y.values

AUC of model 2 = 0.829 while AUC of model3 = 0.782.
Model 2 is preferred to model 3 in AUC.

n) Using model 2 with threshold of 0.5, we get

		Actual 0	Actual 1
model	0	40	18
	1	35	157

The profit matrix is

		Actual 0	Actual 1
Predicted	0	0	0
	1	-300	100

Total profit = 35(-300) + 157(100)

= 5200 DM.

o) Sortpred2 ← Sort (pred2, decreasing = TRUE)

This sorts the predicted probability from high to low

The last entry is row 819 with predicted probability of good risk = 0.0253

germancredit [819,]

For this individual, the duration of credit in months is 36.

p) We first obtain the indices of the sorted data as follows:

```
Sortpred2 ← Sort(pred2, decreasing = TRUE,
                 index.return = TRUE)
```

Sortpred2 $x ⇒ gives the sorted values

Sortpred2 $ix ⇒ gives the indices of sorted values

```
test$resp[Sortpred2$ix]
```

As we see from this that there are more people at the top who are truly good creditors as predicted by the model.

```
profitpred2 ← 100(test$resp[Sortpred2$ix])
              -300(1 - test$resp[Sortpred2$ix])
```

Note that this will give 100 if true value = 1 (right prediction)

& −300 if true value = 0 (wrong prediction)

```
Cumulative ← cumsum(profitpred2)
```

This computes the cumulative profits as we go down the list.

```
which.max(cumulative)
```
150

```
max(Cumulative)
```
7800

This implies we can go down to 150 people out of 250 for maximum profit.

9)

Sortpred2[which.max(cumdelric)]

The corresponding probability is 0.7187.

We would use this as a cutoff to credit good & bad risk based on this data.

4)

a) pres ← read.csv("presidential.csv")

str(pres)

table(pres$WIN)

-1	1
11	14

There have been 14 Democrat presidents winners from 1916 to 2012 & 11 Republican winners.

b) sort(table(pres$DEM))

Roosevelt has represented Democrats 4 times in elections

sort(table(pres$REP))

Nixon has represented Republicans 3 times in elections

c) t.test(pres$GOOD[pres$INC==1], pres$GOOD[pres$INC==-1])

The p-value for the two sided t-test is 0.7494. There is no strong evidence to reject the null hypothesis that the number of good quarters when the president is Democrat or Republican is the same.

d) pres$WININC ← as.integer(pres$INC == pres$WIN)

e) table(pres$WININC)

0	1
9	16

16 times the incumbent party won while 9 times the party lost.

f) model1 ← glm(WININC ~ GROWTH, data = pres, family = binomial)

Summary (model1)

$AIC = 30.365 = -2\,LL(\hat{\beta}) + 2(\text{No of parameters})$

$\therefore LL(\hat{\beta}) = \dfrac{30.365 - 2(2)}{-2} = -13.1825$

g) p-value for GROWTH variable $H_0: \beta_1 = 0$ is 0.0613

Hence the GROWTH variable is significant at the 0.1 level.

h) pres$WIN ← as.integer(pres$WIN == 1)

pres$GROWTH ← pres$GROWTH × pres$INC (*)

Note pres$INC == -1 is Republican. If growth is positive then we set it to be negative from (*) as needed.

i) pres$GOOD ← pres$GOOD * pres$INC

ii) model2 ← glm(WIN ~ INC + RUN + GROWTH + DUR + GOOD, data = pres, family = binomial)

Summary(model2)

AIC = 29.406

j) Least significant variables in the model are:

Intercept,	INC	, GOOD
0.941	0.955	0.728

k) model3 ← glm(WIN ~ RUN + GROWTH + DUR - 1, data = pres, family = binomial)

Summary(model3)

AIC = 23.748

l) p-value for $H_0: \beta_{DUR} = 0$ is 0.1007. So we would reject this if the cutoff is 0.1 but given how close it is, one might be ok to leave it in the model.

m) The second model has a lower AIC value. By dropping the variables, with RUN, GROWTH, DUR we also have a more interpretable model. So we would prefer model3. in (k)

n) INC = 1 (Since Democrats are in power)

RUN = 0 (Since Obama did not run)

DUR = 1 (Since Democrats have been in power for 2 consecutive terms)

o) GROWTH = 2

$$P(Dem=1) = \frac{e^{2.0638(0)+0.4690(2)-1.7852(1)}}{1+e^{0.4690(2)-1.7852(1)}}$$

$$= 0.3$$

$$P(Rep=1) = 0.7$$

```
#1a.i)
baseballlarge <- read.csv("baseballlarge.csv")
str(baseballlarge)

#1a.ii)
length(table(baseballlarge$Year))

#1a.iii)
baseballlarge <- subset(baseballlarge, Playoffs == 1)
str(baseballlarge)

#1a.iv)
table(baseballlarge$Year)
table(table(baseballlarge$Year))

#1b)
baseballlarge$NumCompetitors <-
table(baseballlarge$Year)[as.character(baseballlarge$Year)]
table(baseballlarge$NumCompetitors)j

#1c)
baseballlarge$WorldSeries <- as.integer(baseballlarge$RankPlayoffs == 1)
table(baseballlarge$WorldSeries)

#1d)
model1 <- glm(WorldSeries ~ Year, data = baseballlarge, family = binomial)
model2 <- glm(WorldSeries ~ RS, data = baseballlarge, family = binomial)
model3 <- glm(WorldSeries ~ RA, data = baseballlarge, family = binomial)
model4 <- glm(WorldSeries ~ W, data = baseballlarge, family = binomial)
model5 <- glm(WorldSeries ~ OBP, data = baseballlarge, family = binomial)
model6 <- glm(WorldSeries ~ SLG, data = baseballlarge, family = binomial)
model7 <- glm(WorldSeries ~ BA, data = baseballlarge, family = binomial)
model8 <- glm(WorldSeries ~ RankSeason, data = baseballlarge, family = binomial)
model9 <- glm(WorldSeries ~ NumCompetitors, data = baseballlarge, family =
binomial)
baseballlarge$League<-as.integer(baseballlarge$League == "AL")
model10 <- glm(WorldSeries ~ League, data = baseballlarge, family = binomial)
summary(model1)
summary(model2)
summary(model3)
summary(model4)
summary(model5)
summary(model6)
summary(model7)
summary(model8)
summary(model9)
summary(model10)

#1e)
modelsig <- glm(WorldSeries ~ Year + RA + RankSeason + NumCompetitors, data =
baseballlarge, family = binomial)
summary(modelsig)

#1f)
cor(baseballlarge[,c("Year","RA","RankSeason","NumCompetitors")])

#1g)
modelg1 <- glm(WorldSeries~Year+RA,data=baseballlarge,family=binomial)
modelg2 <- glm(WorldSeries~Year+RankSeason,data=baseballlarge,family=binomial)
modelg3 <-
glm(worldSeries~Year+NumCompetitors,data=baseballlarge,family=binomial)
modelg4 <- glm(WorldSeries~RA+RankSeason,data=baseballlarge,family=binomial)
modelg5 <- glm(WorldSeries~RA+NumCompetitors,data=baseballlarge,family=binomial)
modelg6 <-
glm(WorldSeries~RankSeason+NumCompetitors,data=baseballlarge,family=binomial)
summary(modelg1)
summary(modelg2)
summary(modelg3)
```

```
summary(modelg4)
summary(modelg5)
summary(modelg6)

#2a)
Parole <- read.csv("Parole.csv")
str(Parole)

#2b)
table(Parole$Violator)

#2e)
set.seed(144)
library(caTools)
split <- sample.split(Parole$Violator, SplitRatio = 0.7)
train <- subset(Parole, split == TRUE)
test <- subset(Parole, split == FALSE)
split[1:10]
split <- sample.split(Parole$Violator, SplitRatio = 0.7)
train <- subset(Parole, split == TRUE)
test <- subset(Parole, split == FALSE)
split[1:10]
set.seed(200)
split <- sample.split(Parole$Violator, SplitRatio = 0.7)
train <- subset(Parole, split == TRUE)
test <- subset(Parole, split == FALSE)
split[1:10]
set.seed(144)
split <- sample.split(Parole$Violator, SplitRatio = 0.7)
train <- subset(Parole, split == TRUE)
test <- subset(Parole, split == FALSE)
split[1:10]

#2e)
model1 <- glm(Violator~., data = train, family = binomial)
summary(model1)

#2g)
logodds <-
model1$coef[1]+model1$coef[2]*1+model1$coef[3]*1+model1$coef[4]*50+model1$coef[8]
*3+model1$coef[9]*12+model1$coef[12]*1
odds < exp(logodds)

#2h)
pred <- predict(model1, newdata = test, type = "response")
max(pred)

#2i)
table(as.integer(pred > 0.5), test$violator)

#2j)
table(test$Violator)

#2m)
library(ROCR)
predrocr <- prediction(pred, test$Violator)
auc <- performance(predrocr, measure = "auc")@y.values
auc

#3a)
germancredit <- read.csv("germancredit.csv")
set.seed(2016)
library(caTools)
spl <- sample.split(germancredit$resp,0.75)
training <- subset(germancredit,spl==TRUE)
test <- subset(germancredit,spl==FALSE)

#3b)
```

```
model1 <- glm(resp~-1,data=training,family=binomial)
summary(model1)

#3d)
model2 <- glm(resp~.,data=training,family=binomial)
summary(model2)

#3f)
pred2 <- predict(model2,newdata=test,type="response")
table(pred2>=0.5,test$resp)

#3h)
model3 <-
glm(resp~chkacct+dur+hist+amt+sav+instrate+malesingle+age+for.-1,data=training,fa
mily=binomial)
summary(model3)

#3j)
pred3 <- predict(model3,newdata=test,type="response")
table(pred3>=0.5,test$resp)

#3m)
library(ROCR)
predrocr2 <- prediction(pred2, test$resp)
auc2 <- performance(predrocr2, measure = "auc")@y.values
auc2
predrocr3 <- prediction(pred3, test$resp)
auc3 <- performance(predrocr3, measure = "auc")@y.values
auc3

#3o)
sortpred2 <- sort(pred2,decreasing=TRUE)
sortpred2
germancredit[819,]

#3p)
sortpred2 <- sort(pred2,decreasing=TRUE,index.return=TRUE)
sortpred2$x
sortpred2$ix
test$resp[sortpred2$ix]
profitpred2 <- 100*test$resp[sortpred2$ix] -300*(1-test$resp[sortpred2$ix])
cumulative <- cumsum(profitpred2)
max(cumulative)
which.max(cumulative)

#3q)
sortpred2$x[which.max(cumulative)]

#4a)
pres <- read.csv("presidential.csv")
str(pres)
table(pres$WIN)

#4b)
sort(table(pres$DEM))
sort(table(pres$REP))

#4c)
t.test(pres$GOOD[pres$INC==1],pres$GOOD[pres$INC==-1])

#4d)
pres$WININC <- as.integer(pres$INC==pres$WIN)

#4e)
table(pres$WININC)

#4f)
model1 <- glm(WININC~GROWTH,data=pres,family=binomial)
```

```
#4h)
pres$WIN <- as.integer(pres$WIN==1)
pres$GROWTH <- pres$GROWTH*pres$INC

#4i)
pres$GOOD <- pres$GOOD*pres$INC
model2 <- glm(WIN~INC+RUN+GROWTH+DUR+GOOD,data=pres,family=binomial)
summary(model2)

#4k)
model3 <- glm(WIN~RUN+GROWTH+DUR-1,data=pres,family=binomial)
summary(model3)
```