The Analytics Edge

Test your knowledge of Logistic Regression in R – Solutions

Note to all. I have compiled the answers in the following format – for each question, the qualitative or "written" solutions will be provided together with their sub-questions. The R scripts (as well as the console outputs) will be provided *after* each whole question, followed by all the relevant plots.
If I have missed anything in the solutions, or if you have any questions, you may email me at benjamin_tanwj@mymail.sutd.edu.sg. Thank you!

1. In this question, we will use the data in **baseballlarge.csv** to investigate how well we can predict the World Series winner at the beginning of the playoffs. The dataset has the following fields:

    - Team: A code for the name of the team
    - League: The Major League Baseball league the team belongs to, either AL (American League) or NL (National League)
    - Year: The year of the corresponding record
    - Games: The number of games a team played in that year
    - W: The number of regular season wins by the team in that year
    - RS: The number of runs scored by the team in that year
    - RA: The number of runs allowed by the team in that year
    - OBP: The on-base percentage of the team in that year
    - SLG: The slugging percentage of the team in that year
    - BA: The batting average of the team in that year
    - Playoffs: Whether the team made the playoffs in that year (1 for yes, 0 for no)
    - RankSeason: Among the playoff teams in that year, the ranking of their regular season records (1 is best)
    - RankPlayoffs: Among the playoff teams in that year, how well they fared in the playoffs. The team winning the World Series gets a RankPlayoffs of 1.

    (a) Each row in the baseball dataset represents a team in a particular year. Read the data into a dataframe called "baseballlarge".

    i. How many team/year pairs are there in the whole dataset?

    ii. Though the dataset contains data from 1962 until 2012, we removed several years with shorter-than-usual seasons. Using the table() function, identify the total number of years included in this dataset.

iii. Because we're only analyzing teams that made the playoffs, use the subset() function to create a smaller data frame limited to teams that made the playoffs. Your subsetted data frame should still be called "baseballlarge". How many team/year pairs are included in the new dataset?

iv. Through the years, different numbers of teams have been invited to the playoffs. Find the different number of teams making the playoffs across the seasons.

Solution.

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

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

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

iv. See R scripts.

(b) It's much harder to win the World Series if there are 10 teams competing for the championship versus just two. Therefore, we will add the predictor variable NumCompetitors to the data frame. NumCompetitors will contain the number of total teams making the playoffs in the year of a particular team/year pair. For instance, NumCompetitors should be 2 for the 1962 New York Yankees, but it should be 8 for the 1998 Boston Red Sox. We want to look up the number of teams in the playoffs for each team/year pair in the dataset, and store it as a new variable named NumCompetitors in the data frame. Do this. How many playoff team/year pairs are there in the dataset from years where 8 teams were invited to the playoffs?

Solution. See R scripts. Note that to obtain a particular entry from the table, we need to call it with `table(baseballlarge$year)["1962"]`
`table(baseballlarge$NumCompetitors)` 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) In this problem, we seek to predict whether a team won the World Series; in our dataset this is denoted with a RankPlayoffs value of 1. Add a variable named WorldSeries to the data frame that takes value 1 if a team won the World Series in the indicated year and a 0 otherwise. How many observations do we have in our dataset where a team did NOT win the World Series?

Solution. There are 197 team/year pairs in the dataset who did not win the World Series.

(d) When we're not sure which of our variables are useful in predicting a particular outcome, it's often helpful to build simple models, which are models that predict the outcome using a single independent variable. Which of the variables is a significant predictor of the

WorldSeries variable in a logistic regression model? To determine significance, remember to look at the stars in the summary output of the model. We'll define an independent variable as significant if there is at least one star at the end of the coefficients row for that variable (this is equivalent to the probability column having a value smaller than 0.05). Note that you have to build multiple models (Year, RS, RA, W, OBP, SLG, BA, RankSeason, NumCompetitors, League) to answer this question (you can code the League variable as a categorical variable). Use the dataframe baseballlarge to build the models.

Solution. See R scripts. Note that for the *League* variable we do `baseballlarge$League <- as.integer(baseballlarge$League == "AL")`. In here, AL is stored as 1 and NL as 0. Since there are only 2 categories (unordered), this approach makes sense of handling the independent variable.

From the results, we find that the *Year, RA, RankSeason, NumCompetitors* variables are significant. W and SLG just misses out with p-values of 0.0577 and 0.0504 respectively.

(e) In this question, we will consider multivariate models that combine the variables we found to be significant in (d). Build a model using all of the variables that you found to be significant in (d). How many variables are significant in the combined model?

Solution. Note that while we will include *Year, RA, RankSeason, NumCompetitors*, it also makes sense to include W and SLG since they have p-values close to 0.05. However, in this new multivariate model, none of the variables are significant.

(f) Often, variables that were significant in single variable models are no longer significant in multivariate analysis due to correlation between the variables. Are there any such variables in this example? Which of the variable pairs have a high degree of correlation (a correlation greater than 0.8 or less than -0.8)?

Solution. In this example, the year and number of competitors has a high correlation.

(g) Build all of the two variable models from (f). Together with the models from (d), you should have different logistic regression models. Which model has the best AIC value (the minimum AIC value)?

Solution. The smallest AIC value corresponds to model 9: *WorldSeries NumCompetitors*.

(h) Comment on your results.

Solution. Overall it seems the winning of playoffs has a large role of luck and other season performances do not play a major role.

R Scripts.

```
> #1a.i)
> baseballlarge <- read.csv("baseballlarge.csv")
> str(baseballlarge)
'data.frame': 1232 obs. of  13 variables:
 $ Team        : Factor w/ 39 levels "ANA","ARI","ATL",..: 2 3 4 5 7 8 9 10 11 12 ...
 $ League      : Factor w/ 2 levels "AL","NL": 2 2 1 1 2 1 2 1 2 1 ...
 $ Year        : int  2012 2012 2012 2012 2012 2012 2012 2012 2012 2012 ...
 $ RS          : int  734 700 712 734 613 748 669 667 758 726 ...
 $ RA          : int  688 600 705 806 759 676 588 845 890 670 ...
 $ W           : int  81 94 93 69 61 85 97 68 64 88 ...
 $ OBP         : num  0.328 0.32 0.311 0.315 0.302 0.318 0.315 0.324 0.33 0.335 ...
 $ SLG         : num  0.418 0.389 0.417 0.415 0.378 0.422 0.411 0.381 0.436 0.422 ...
 $ BA          : num  0.259 0.247 0.247 0.26 0.24 0.255 0.251 0.251 0.274 0.268 ...
 $ Playoffs    : int  0 1 1 0 0 0 1 0 0 1 ...
 $ RankSeason  : int  NA 4 5 NA NA NA 2 NA NA 6 ...
 $ RankPlayoffs: int  NA 5 4 NA NA NA 4 NA NA 2 ...
 $ Games       : int  162 162 162 162 162 162 162 162 162 162 ...
> #1a.ii)
> length(table(baseballlarge$Year))
[1] 47
> #1a.iii)
> baseballlarge <- subset(baseballlarge, Playoffs == 1)
> str(baseballlarge)
'data.frame': 244 obs. of  13 variables:
 $ Team        : Factor w/ 39 levels "ANA","ARI","ATL",..: 3 4 9 12 25 26 32 33 36 39 ...
 $ League      : Factor w/ 2 levels "AL","NL": 2 1 2 1 1 1 2 2 1 2 ...
 $ Year        : int  2012 2012 2012 2012 2012 2012 2012 2012 2012 2012 ...
 $ RS          : int  700 712 669 726 804 713 718 765 808 731 ...
 $ RA          : int  600 705 588 670 668 614 649 648 707 594 ...
 $ W           : int  94 93 97 88 95 94 94 88 93 98 ...
 $ OBP         : num  0.32 0.311 0.315 0.335 0.337 0.31 0.327 0.338 0.334 0.322 ...
 $ SLG         : num  0.389 0.417 0.411 0.422 0.453 0.404 0.397 0.421 0.446 0.428 ...
 $ BA          : num  0.247 0.247 0.251 0.268 0.265 0.238 0.269 0.271 0.273 0.261 ...
 $ Playoffs    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ RankSeason  : int  4 5 2 6 3 4 4 6 5 1 ...
 $ RankPlayoffs: int  5 4 4 2 3 4 1 3 5 4 ...
 $ Games       : int  162 162 162 162 162 162 162 162 162 162 ...
> #1a.iv)
> table(baseballlarge$Year)
```

```
1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1973 1974 1975 1976 1977 1978 1979 1980
   2    2    2    2    2    2    2    4    4    4    4    4    4    4    4    4    4    4
1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1996 1997 1998 1999 2000 2001
   4    4    4    4    4    4    4    4    4    4    4    4    8    8    8    8    8    8
2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
   8    8    8    8    8    8    8    8    8    8   10
> table(table(baseballlarge$Year))

 2  4  8 10
 7 23 16  1


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

  2   4   8  10
 14  92 128  10


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

  0   1
197  47


> #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)

Call:
glm(formula = WorldSeries ~ Year, family = binomial, data = baseballlarge)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0297  -0.6797  -0.5435  -0.4648   2.1504

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 72.23602   22.64409    3.19  0.00142 **
Year        -0.03700    0.01138   -3.25  0.00115 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 228.35  on 242  degrees of freedom
AIC: 232.35

Number of Fisher Scoring iterations: 4

> summary(model2)

Call:
glm(formula = WorldSeries ~ RS, family = binomial, data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8254  -0.6819  -0.6363  -0.5561   2.0308

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.661226   1.636494   0.404    0.686
RS          -0.002681   0.002098  -1.278    0.201

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 237.45  on 242  degrees of freedom
AIC: 241.45

Number of Fisher Scoring iterations: 4

> summary(model3)
```

```
Call:
glm(formula = WorldSeries ~ RA, family = binomial, data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9749  -0.6883  -0.6118  -0.4746   2.1577

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.888174   1.483831   1.272   0.2032
RA          -0.005053   0.002273  -2.223   0.0262 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 233.88  on 242  degrees of freedom
AIC: 237.88

Number of Fisher Scoring iterations: 4

> summary(model4)

Call:
glm(formula = WorldSeries ~ W, family = binomial, data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0623  -0.6777  -0.6117  -0.5367   2.1254

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.85568    2.87620  -2.384   0.0171 *
W            0.05671    0.02988   1.898   0.0577 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 235.51  on 242  degrees of freedom
AIC: 239.51
```

```
Number of Fisher Scoring iterations: 4

> summary(model5)

Call:
glm(formula = WorldSeries ~ OBP, family = binomial, data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8071  -0.6749  -0.6365  -0.5797   1.9753

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    2.741      3.989   0.687    0.492
OBP          -12.402     11.865  -1.045    0.296

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 238.02  on 242  degrees of freedom
AIC: 242.02

Number of Fisher Scoring iterations: 4

> summary(model6)

Call:
glm(formula = WorldSeries ~ SLG, family = binomial, data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9498  -0.6953  -0.6088  -0.5197   2.1136

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    3.200      2.358   1.357   0.1748
SLG          -11.130      5.689  -1.956   0.0504 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 235.23  on 242  degrees of freedom
```

```
AIC: 239.23

Number of Fisher Scoring iterations: 4

> summary(model7)

Call:
glm(formula = WorldSeries ~ BA, family = binomial, data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.6797  -0.6592  -0.6513  -0.6389   1.8431

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.6392     3.8988  -0.164    0.870
BA           -2.9765    14.6123  -0.204    0.839

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 239.08  on 242  degrees of freedom
AIC: 243.08

Number of Fisher Scoring iterations: 4

> summary(model8)

Call:
glm(formula = WorldSeries ~ RankSeason, family = binomial, data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7805  -0.7131  -0.5918  -0.4882   2.1781

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.8256     0.3268  -2.527   0.0115 *
RankSeason   -0.2069     0.1027  -2.016   0.0438 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 234.75  on 242  degrees of freedom
AIC: 238.75

Number of Fisher Scoring iterations: 4

> summary(model9)

Call:
glm(formula = WorldSeries ~ NumCompetitors, family = binomial,
    data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9871  -0.8017  -0.5089  -0.5089   2.2643

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.03868    0.43750   0.088 0.929559
NumCompetitors -0.25220    0.07422  -3.398 0.000678 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 226.96  on 242  degrees of freedom
AIC: 230.96

Number of Fisher Scoring iterations: 4

> summary(model10)

Call:
glm(formula = WorldSeries ~ League, family = binomial, data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.6772  -0.6772  -0.6306  -0.6306   1.8509

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.5141     0.2355  -6.430 1.28e-10 ***
League        0.1583     0.3252   0.487    0.626
```

```
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 238.88  on 242  degrees of freedom
AIC: 242.88

Number of Fisher Scoring iterations: 4


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

Call:
glm(formula = WorldSeries ~ Year + RA + RankSeason + NumCompetitors,
    family = binomial, data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0336  -0.7689  -0.5139  -0.4583   2.2195

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    12.5874376 53.6474210   0.235    0.814
Year           -0.0061425  0.0274665  -0.224    0.823
RA             -0.0008238  0.0027391  -0.301    0.764
RankSeason     -0.0685046  0.1203459  -0.569    0.569
NumCompetitors -0.1794264  0.1815933  -0.988    0.323

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 226.37  on 239  degrees of freedom
AIC: 236.37

Number of Fisher Scoring iterations: 4


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

```
                   Year        RA RankSeason NumCompetitors
Year           1.0000000 0.4762422  0.3852191      0.9139548
RA             0.4762422 1.0000000  0.3991413      0.5136769
RankSeason     0.3852191 0.3991413  1.0000000      0.4247393
NumCompetitors 0.9139548 0.5136769  0.4247393      1.0000000


> #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)

Call:
glm(formula = WorldSeries ~ Year + RA, family = binomial, data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0402  -0.6878  -0.5298  -0.4785   2.1370

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 63.610741  25.654830   2.479   0.0132 *
Year        -0.032084   0.013323  -2.408   0.0160 *
RA          -0.001766   0.002585  -0.683   0.4945
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 227.88  on 241  degrees of freedom
AIC: 233.88

Number of Fisher Scoring iterations: 4

> summary(modelg2)

Call:
glm(formula = WorldSeries ~ Year + RankSeason, family = binomial,
    data = baseballlarge)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0560  -0.6957  -0.5379  -0.4528   2.2673

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 63.64855   24.37063   2.612  0.00901 **
Year        -0.03254    0.01231  -2.643  0.00822 **
RankSeason  -0.10064    0.11352  -0.887  0.37534
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 227.55  on 241  degrees of freedom
AIC: 233.55

Number of Fisher Scoring iterations: 4

> summary(modelg3)

Call:
glm(formula = WorldSeries ~ Year + NumCompetitors, family = binomial,
    data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0050  -0.7823  -0.5115  -0.4970   2.2552

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    13.350467  53.481896   0.250    0.803
Year           -0.006802   0.027328  -0.249    0.803
NumCompetitors -0.212610   0.175520  -1.211    0.226

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 226.90  on 241  degrees of freedom
AIC: 232.9

Number of Fisher Scoring iterations: 4
```

```
> summary(modelg4)

Call:
glm(formula = WorldSeries ~ RA + RankSeason, family = binomial,
    data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9374  -0.6933  -0.5936  -0.4564   2.1979

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.487461   1.506143   0.988    0.323
RA          -0.003815   0.002441  -1.563    0.118
RankSeason  -0.140824   0.110908  -1.270    0.204

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 232.22  on 241  degrees of freedom
AIC: 238.22

Number of Fisher Scoring iterations: 4

> summary(modelg5)

Call:
glm(formula = WorldSeries ~ RA + NumCompetitors, family = binomial,
    data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0433  -0.7826  -0.5133  -0.4701   2.2208

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.716895   1.528736   0.469  0.63911
RA             -0.001233   0.002661  -0.463  0.64313
NumCompetitors -0.229385   0.088399  -2.595  0.00946 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 226.74  on 241  degrees of freedom
AIC: 232.74

Number of Fisher Scoring iterations: 4

> summary(modelg6)

Call:
glm(formula = WorldSeries ~ RankSeason + NumCompetitors, family = binomial,
    data = baseballlarge)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0090  -0.7592  -0.5204  -0.4501   2.2562

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     0.12277    0.45737   0.268  0.78837
RankSeason     -0.07697    0.11711  -0.657  0.51102
NumCompetitors -0.22784    0.08201  -2.778  0.00546 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 239.12  on 243  degrees of freedom
Residual deviance: 226.52  on 241  degrees of freedom
AIC: 232.52

Number of Fisher Scoring iterations: 4
```

2. In this question, we will build on the Parole.csv dataset from the first week to build and validate a model that predicts if an inmate will violate the terms of his or her parole. Such a model could be useful to a parole board when deciding to approve or deny an application for parole.

   (a) Load the dataset Parole.csv into a data frame called Parole. How many parolees are contained in the dataset?

   Solution. There are a total of 675 parolees in this date.

   (b) How many of the parolees in the dataset violated the terms of their parole?

   Solution. 78 of the parolees violated their terms of parole.

   (c) Factor variables are variables that take on a discrete set of values and can be either unordered or ordered. Names of countries indexed by levels is an example of an unordered factor because there isn't any natural ordering between the levels. An ordered factor has a natural ordering between the levels (an example would be the classifications "large", "medium" and "small"). Which variables in this dataset are unordered factors with at least three levels? To deal with unordered factors in a regression model, the standard practice is to define one level as the "reference level" and create a binary variable for each of the remaining levels. In this way, a factor with n levels is replaced by n-1 binary variables. We will see this in question (e).

   Solution. In this data, *State* and *Crime* are unordered factor variables with at least 3 variables.

   (d) To ensure consistent training/testing set splits, run the following 5 lines of code (do not include the line numbers at the beginning):

   (1) > set.seed(144)
   (2) > library(caTools)
   (3) > split <− sample.split(Parole$Violator, SplitRatio = 0.7)
   (4) > train <− subset(Parole, split == TRUE)
   (5) > test <− subset(Parole, split == FALSE)

   Roughly what proportion of parolees have been allocated to the training and testing sets? Now, suppose you re-ran lines (1)-(5) again. What would you expect?

   - The exact same training/testing set split as the first execution of (1)-(5)
   - A different training/testing set split from the first execution of (1)-(5)

   If you instead ONLY re-ran lines (3)-(5), what would you expect?

   - The exact same training/testing set split as the first execution of (1)-(5)

- A different training/testing set split from the first execution of (1)-(5)

If you instead called set.seed() with a different number and then re-ran lines (3)-(5), what would you expect?

- The exact same training/testing set split as the first execution of (1)-(5)
- A different training/testing set split from the first execution of (1)-(5)

Solution. In the split, roughly 70% of the parolees have been assigned to the training and 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, and this is because we set the seed (random number generator) to be the same. It follows naturally that if we only run commands $(3)-(5)$ without setting a seed, or setting a seed to a different number from 144, we would get a different training/test split.

(e) If you tested other training/testing set splits in the previous section, please re-run the original 5 lines of code to obtain the original split. Using glm, train a logistic regression model on the training set. Your dependent variable is "Violator", and you should use all the other variables as independent variables. What variables are significant in this model? Significant variables should have a least one star, or should have a p-value less than 0.05.

Solution. The significant variables at the 0.05 level are *RaceWhite* (race of parolee), *StateVirginia* and *MultipleOffenses*.

(f) What can we say based on the coefficient of the MultipleOffenses variable?

- Our model predicts that parolees who committed multiple offenses have 1.61 times higher odds of being a violator than the average parolee.
- Our model predicts that a parolee who committed multiple offenses has 1.61 times higher odds of being a violator than a parolee who did not commit multiple offenses but is otherwise identical.
- Our model predicts that parolees who committed multiple offenses have 5.01 times higher odds of being a violator than the average parolee.
- Our model predicts that a parolee who committed multiple offenses has 5.01 times higher odds of being a violator than a parolee who did not commit multiple offenses but is otherwise identical.

Solution. From the logistic regression result, we get

$$\beta_{MultipleOffenses} = 1.61.$$

Everything else being equal, we have

$$Odds = \frac{P(Y=1)}{P(Y=0)} = e^{\beta_{MultipleOffenses} \cdot MultipleOffenses}.$$

If a person did not commit multiple offenses, this number *MultipleOffenses* $= 0$, else 1. This means that the odds is equal to 5.01, and represents the odds of a parolee to be a violator with multiple offenses, compared to a person who did not commit multiple offenses but is otherwise identical. Statement (4) is the appropriate one.

(g) Consider a parolee who is male, of white race, aged 50 years at prison release, from Kentucky, served 3 months, had a maximum sentence of 12 months, did not commit multiple offenses, and committed a larceny. According to the model, what are the odds this individual is a violator? According to the model, what is the probability this individual is a violator?

Solution. The log odds of the given individual being a violator is

$$\beta^T x = \begin{pmatrix} \beta_0 \\ \beta_{Male} \\ \beta_{RaceWhite} \\ \beta_{Age} \\ \beta_{TimeServed} \\ \beta_{MaxSentence} \\ \beta_{MultipleOffenses} \\ \beta_{CrimeLarceny} \end{pmatrix}^T \begin{pmatrix} 1 \\ Male \\ RaceWhite \\ Age \\ TimeServed \\ MaxSentence \\ MultipleOffenses \\ CrimeLarceny \end{pmatrix}$$

$$= \begin{pmatrix} -2.03 \\ 0.38 \\ -0.88 \\ -0.0017 \\ -0.12 \\ 0.08 \\ 1.66 \\ 0.695 \end{pmatrix}^T \begin{pmatrix} 1 \\ 1 \\ 1 \\ 50 \\ 3 \\ 12 \\ 0 \\ 1 \end{pmatrix} = -1.25$$

This means that the odds are given as $e^{-1.25} = 0.28$.

(h) Use the predict() function to obtain the model's predicted probabilities for parolees in the test set. What is the maximum predicted probability of a violation?

Solution. The maximum predicted probability of violation is given as 0.9707.

(i) In the following questions, evaluate the model's predictions on the test set using a threshold of 0.5. What is the model's sensitivity? What is the model's specificity? What is the model's accuracy?

Solution.

$$Sensitivity = TPR = \frac{12}{12 + 11} = 0.521.$$

$$Specificity = TNR = \frac{167}{167 + 12} = 0.932.$$

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

(j) What is the accuracy of a simple model that predicts that every parolee is a non-violator?

Solution. If the model predicts everyone as a non-violator:

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

We get the same accuracy as the previous model.

(k) Consider a parole board using the model to predict whether parolees will be violators or not. The job of a parole board is to make sure that a prisoner is ready to be released into free society, and therefore parole boards tend to be particularily concerned with releasing prisoners who will violate their parole. Which of the following most likely describes their preferences and best course of action?

- The board assigns more cost to a false negative than a false positive, and should therefore use a logistic regression cutoff higher than 0.5.
- The board assigns more cost to a false negative than a false positive, and should therefore use a logistic regression cutoff less than 0.5.
- The board assigns equal cost to a false positive and a false negative, and should therefore use a logistic regression cutoff equal to 0.5.
- The board assigns more cost to a false positive than a false negative, and should therefore use a logistic regression cutoff higher than 0.5.
- The board assigns more cost to a false positive than a false negative, and should therefore use a logistic regression cutoff less than 0.5.

Solution. 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 false negatives than false positives, and should use a cutoff less than 0.5.

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

(l) Which of the following is the most accurate assessment of the value of the logistic regression model with a cutoff 0.5 to a parole board, based on the model's accuracy as compared to the simple baseline model?

- The model is of limited value to the board because it cannot outperform a simple baseline, and using a different logistic regression cutoff is unlikely to improve the model's value.
- The model is of limited value to the board because it cannot outperform a simple baseline, and using a different logistic regression cutoff is likely to improve the model's value.
- The model is likely of value to the board, and using a different logistic regression cutoff is unlikely to improve the model's value.
- The model is likely of value to the board, and using a different logistic regression cutoff is likely to improve the model's value.

Solution. The model is of likely value to the board since it can provide a better characterisation 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) Using the ROCR package, what is the AUC value for the model?

Solution. $AUC = 0.894$.

(n) Describe the meaning of AUC in this context.

- The probability the model can correctly differentiate between a randomly selected parole violator and a randomly selected parole non-violator.
- The model's accuracy at logistic regression cutoff of 0.5.
- The model's accuracy at the logistic regression cutoff at which it is most accurate.

Solution. The AUC can be interpreted as the probability that the model can correctly differentate between a randomly-selected parole violator, and a randomly-selected parole non-violator.

(o) Our goal has been to predict the outcome of a parole decision, and we used a publicly available dataset of parole releases for predictions. In this final problem, we'll evaluate a potential source of bias associated with our analysis. It is always important to evaluate a dataset for possible sources of bias. The dataset contains all individuals released from parole in 2004, either due to completing their parole term or violating the terms of their parole. However, it does not contain parolees who neither violated their parole nor completed their term in 2004, causing non-violators to be underrepresented. This is called “selection bias” or “selecting on the dependent variable,” because only a subset of all relevant parolees were included in our analysis, based on our dependent variable in this analysis (parole violation). How could we improve our dataset to best address selection bias?

- There is no way to address this form of biasing.
- We should use the current dataset, expanded to include the missing parolees. Each added parolee should be labeled with Violator=0, because they have not yet had a violation.
- We should use the current dataset, expanded to include the missing parolees. Each added parolee should be labeled with Violator=NA, because the true outcome has not been observed for these individuals.
- We should use a dataset tracking a group of parolees from the start of their parole until either they violated parole or they completed their term.

Solution. Option 2 does not capture the true outcome of parolees since they are still either in jail, or not violated thus far. Option 3 does 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 dataset requires more effort to gather.

R Scripts.

```
> #2a)
> Parole <- read.csv("Parole.csv")
> str(Parole)
'data.frame': 675 obs. of  9 variables:
 $ Male            : int  1 0 1 1 1 1 1 0 0 1 ...
 $ RaceWhite       : int  1 1 0 1 0 0 1 1 1 0 ...
 $ Age             : num  33.2 39.7 29.5 22.4 21.6 46.7 31 24.6 32.6 29.1 ...
 $ State           : Factor w/ 4 levels "Kentucky","Louisiana",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ TimeServed      : num  5.5 5.4 5.6 5.7 5.4 6 6 4.8 4.5 4.7 ...
 $ MaxSentence     : int  18 12 12 18 12 18 18 12 13 12 ...
 $ MultipleOffenses: int  0 0 0 0 0 0 0 0 0 0 ...
 $ Crime           : Factor w/ 4 levels "Driving","Drugs",..: 1 2 2 4 4 1 2 4 2 3 ...
 $ Violator        : int  0 0 0 0 0 0 0 0 0 0 ...

> #2b)
> table(Parole$Violator)

  0   1
597  78
```

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

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

Call:
glm(formula = Violator ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7041  -0.4236  -0.2719  -0.1690   2.8375

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -2.0361809  1.4474831  -1.407   0.1595
Male              0.3869904  0.4379613   0.884   0.3769
RaceWhite        -0.8867192  0.3950660  -2.244   0.0248 *
Age              -0.0001756  0.0160852  -0.011   0.9913
StateLouisiana    0.3916790  0.5719679   0.685   0.4935
StateOther       -0.4433007  0.4816619  -0.920   0.3574
```

```
StateVirginia    -3.8400884  0.6904894  -5.561 2.68e-08 ***
TimeServed       -0.1238867  0.1204230  -1.029   0.3036
MaxSentence       0.0802954  0.0553747   1.450   0.1470
MultipleOffenses  1.6119919  0.3853050   4.184 2.87e-05 ***
CrimeDrugs       -0.2663428  0.6412857  -0.415   0.6779
CrimeLarceny      0.6954770  0.6714835   1.036   0.3003
CrimeOther        0.0117627  0.5713035   0.021   0.9836
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 340.04  on 472  degrees of freedom
Residual deviance: 251.48  on 460  degrees of freedom
AIC: 277.48

Number of Fisher Scoring iterations: 6


> #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)
> logodds
(Intercept)
  -1.257329
> odds
(Intercept)
  0.2844126

> #2h)
> pred <- predict(model1, newdata = test, type = "response")
> max(pred)
[1] 0.9072791

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

      0   1
  0 167  11
  1  12  12

> #2j)
> table(test$Violator)
```

```
  0   1 
179  23 

> #2m)
> library(ROCR)
> predrocr <- prediction(pred, test$Violator)
> auc <- performance(predrocr, measure = "auc")@y.values
> auc
[[1]]
[1] 0.8945834
```

3. Credit scoring rules are used to determine if a new applicant should be classified as a good credit risk or a bad credit risk, based on values for one or more of the predictor variables. Lenders such as banks and credit card companies use credit scores to determine if money should be lent to consumers. The file **germandcredit.csv** contains data on 1000 past credit applicants from Germany. This data is obtained from the UCI Machine Learning Repository. In this question, we want to develop a model that may be used to determine if new applicants present a good credit risk or a bad credit risk. The dataset contains the following variables:

   - **chkacct**: Status of existing checking account. Variable is coded as:
     0: < 0 DM (Deutsche Mark)
     1: $0 \leq \ldots < 200$ DM
     2: ≥ 200 DM/salary assignments for at least 1 year
     3: no checking account
   - **dur**: Duration of credit in months
   - **hist**: Credit history. Variable is coded as:
     0: no credits taken
     1: all credits at this bank paid back duly
     2: existing credits paid back duly till now
     3: delay in paying off in the past
     4: critical account
   - **newcar**: Purpose of credit (new car). Variable is coded as: 0: No, 1: Yes
   - **usedcar**: Purpose of credit (used car). Variable is coded as: 0: No, 1: Yes
   - **furn**: Purpose of credit (used furniture/equipment). Variable is coded as: 0: No, 1: Yes
   - **radiotv**: Purpose of credit (radio/television). Variable is coded as: 0: No, 1: Yes
   - **educ**: Purpose of credit (education). Variable is coded as: 0: No, 1: Yes
   - **retrain**: Purpose of credit (retraining). Variable is coded as: 0: No, 1: Yes
   - **amt**: Credit amount
   - **sav**: Average balance in savings account. Variable is coded as:
     0: < 100 DM
     1: $100 \leq \ldots < 500$ DM
     2: $500 \leq \ldots < 1000$ DM
     3: ≥ 1000 DM
     4: unknown/ no savings account
   - **emp**: Present employment since. Variable is coded as:
     0: unemployed
     1: < 1 year
     2: $1 \leq \ldots < 4$ years
     3: $4 \leq \ldots < 7$ years
     4: ≥ 7 years

- **instrate**: Installment rate in percentage of disposable income
- **malediv**: Applicant is male and divorced. Variable is coded as: 0: No, 1: Yes
- **malesingle**: Applicant is male and single. Variable is coded as: 0: No, 1: Yes
- **malemarwid**: Applicant is male and married or a widower. Variable is coded as: 0: No, 1: Yes
- **coapp**: Applicant has a co-applicant. Variable is coded as: 0: No, 1: Yes
- **guar**: Applicant has a guarantor. Variable is coded as: 0: No, 1: Yes
- **presres**: Present resident since in years. Variable is coded as:
  0: ≤ 1 year
  1: $1 < \ldots \leq 2$ years
  2: $2 < \ldots \leq 3$ years
  3: > 4 years
- **realest**: Applicant owns real estate. Variable is coded as: 0: No, 1: Yes
- **propnone**: Applicant owns no property (or unknown). Variable is coded as: 0: No, 1: Yes
- **age**: Age in years
- **other**: Applicant has other installment plan credit. Variable is coded as: 0: No, 1: Yes
- **rent**: Applicant rents. Variable is coded as: 0: No, 1: Yes
- **ownres**: Applicant owns residence. Variable is coded as: 0: No, 1: Yes
- **numcred**: Number of existing credits at this bank
- **job**: Nature of job. Variable is coded as:
  0: unemployed/unskilled - non-resident
  1: unskilled - resident
  2: skilled employee/official
  3: management/self-employed/highly qualified employee/officer
- **numdep**: Number of people for whom liable to provide maintenance
- **tel**: Applicant has phone in his or her name. Variable is coded as: 0: No, 1: Yes
- **foreign**: Foreign worker. Variable is coded as: 0: No, 1: Yes
- **resp**: Credit rating is good. Variable is coded as: 0: No, 1: Yes

(a) Read the data into the dataframe **germancredit**. We are interested in predicting the **resp** variable. Obtain a random training/test set split with:
> set.seed(2016)
> library(caTools)
> spl <- sample.split(germancredit$resp, 0.75)
Split the data frame into a training data frame called "training" using the observations for which spl is TRUE and a test data frame called "test" using the observations for which spl is FALSE. Why do we use the sample.split() function to split into a training and testing set?

- It is the only method in R to randomly split the data.
- It balances the independent variables between the training and testing sets.
- It balances the dependent variable between the training and testing sets.

Select the best option.

Solution. See R scripts. The reason for splitting the dataset in such a way is to ensure that the dependent variable is balanced between the training and test sets.

(b) We start with the simplest logistic regression model to predict credit risk in the training set using no predictor variables except the constant (intercept). Write down the fitted model.

Solution. 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) Provide a precise mathematical relationship between the estimated coefficient and the fraction of respondents with a good credit rating in the training set.

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

(d) We now develop a logistic regression model to predict credit card default using all the possible predictor variables. Identify all variable that are significant at the 10% level.

Solution. The variables that are significant at the 10% level are *chckacct, dur, hist, amt, sav, instrate, malesingle, age, for.*

(e) What is the log likelihood value for this model?

Solution.

$$LL(\hat{\beta}) = -\frac{1}{2} \textit{Residual Deviance} = -344.775.$$

(f) Compute the confusion matrix on the test set. For the logistic regression model use a threshold of 0.5.

Solution. See R scripts.

(g) What is the accuracy of the model?

Solution.
$$Accuracy = \frac{157 + 40}{157 + 40 + 18 + 35} = 0.788.$$

(h) Redo the logistic regression model to predict credit risk using only the predictor variables that were significant at the 10% level in (d). What is the AIC value for this model?

Solution. AIC of model 3 is 750.24.

(i) Based on the AIC, which model is preferable?

Solution. AIC of model 2 is 751.55. Model 3 is preferred to model 2.

(j) Compute the confusion matrix on the test set for the model in (h). For the logistic regression model use a threshold of 0.5.

Solution. See R scripts.

(k) Based on the fraction of people who are predicted as good credit risk but are actually bad credit risk in the test set, which model is preferable?

Solution. In model 2, the fraction of people who are predicted as good risk, but are actually bad risk (Note that this is a Type I error!) is $\frac{35}{35+40} = 0.4667$. In model 3, this fraction is $\frac{42}{42+33} = 0.56$. Clearly, model 2 is preferred if we use this metric for comparison.

(l) Based on the fraction of people who are predicted as bad credit risk but are actually good credit risk in the test set, which model is preferable?

Solution. In model 2, the fraction of people who are predicted as bad risk, but are actually good risk (Note that this is a Type II error!) is $\frac{18}{18+157} = 0.102$. In model 3, this fraction is $\frac{15}{15+160} = 0.0857$. Model 3 would be preferred if we use this metric.

(m) Based on the area under the curve in the test set, which model is preferable?

Solution. AUC for model 2 is 0.829, whereas that of model 3 is 0.782. Model 2 would be preferred to model 3.

(n) From this point onwards, we use the model with all the predictor variables included. We now consider a more sophisticated way to evaluate the consequence of misclassification. The consequences of misclassification by the credit company is assessed as follows: the costs of incorrectly saying an applicant is a good credit risk is 300 DM while the profit of correctly saying an applicant is a good credit risk is 100 DM. In terms of profit this can be considered in terms of a table as follows:

	Actual Bad	Actual Good
Predicted Bad	0	0
Predicted Good	-300 DM	100 DM

What is the total profit incurred by the credit company on the test set?

Solution. See (f) for confusion matrix. The total profit would be $35(-300) + 157(100) = 5200DM$.

(o) To see if we can improve the performance by changing the threshold, we will use the predicted probability of credit risk from the logistic regression as a basis by selecting the good credit risks first, followed by poorer risk applicants. Sort the test set on the predicted probability of good credit risk from high to low (Hint: You can use the sort command). What is the duration of credit in months for the individual with the lowest predicted probability of good credit risk?

Solution. `sortpred2 <- sort(pred2, decreasing=TRUE)` sorts the predicted probability from high to low. The last entry is row 819 with predicted probability of good risk azt 0.0253. For this individual, the duration of credit in months is 36.

(p) For each observation in the sorted test set, calculate the actual profit of providing credit (use the table in (n)). Compute the net profit by adding a new variable that captures the cumulative profit. How many far down the test set do you need to go in order to get the maximum profit? (Hint. You can use the index from the index.return argument in the sort function and use the cumsum function)

Solution. We first obtain the indices of the sorted dates by running the command

`sortpred2 <- sort(pred2, decreasing=TRUE, index.return=TRUE).`

`sortpred2$x` gives the sorted values, and `sortpred2$ix` returns the indices of the sorted values. It seems that there are more people at the top who are truly good creditors as predicted by the model. The maximum profit is given as 7800, with argmax at the 150th person.

(q) If the logistic regression model from (p) is scored to future applicants, what "probability of good credit risk" cutoff should be used in extending credit?

Solution. The corresponding probability is 0.7187. We would use this as a cutoff to credit goal and bad risk based on this data.

R Scripts.

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

  0   1
225 525
> table(test$resp)

  0   1
 75 175

> #3b)
> model1 <- glm(resp~-1,data=training,family=binomial)
> summary(model1)

Call:
glm(formula = resp ~ -1, family = binomial, data = training)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.177  -1.177   1.177   1.177   1.177

No Coefficients

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1039.7  on 750  degrees of freedom
Residual deviance: 1039.7  on 750  degrees of freedom
```

```
AIC: 1039.7

Number of Fisher Scoring iterations: 0


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

Call:
glm(formula = resp ~ ., family = binomial, data = training)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6850  -0.7577   0.4041   0.6919   1.9701

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  8.764e-01  1.028e+00   0.853 0.393872
chkacct      5.652e-01  8.470e-02   6.674 2.50e-11 ***
dur         -2.053e-02  1.024e-02  -2.005 0.045012 *
hist         4.764e-01  1.008e-01   4.727 2.28e-06 ***
newcar      -6.703e-01  4.623e-01  -1.450 0.147105
usedcar      7.830e-01  5.755e-01   1.361 0.173630
furn        -5.487e-02  4.780e-01  -0.115 0.908623
radiotv      8.473e-02  4.620e-01   0.183 0.854492
educ        -8.672e-01  5.896e-01  -1.471 0.141351
retrain      2.284e-01  5.271e-01   0.433 0.664745
amt         -1.525e-04  4.976e-05  -3.064 0.002186 **
sav          2.379e-01  7.006e-02   3.395 0.000686 ***
emp          5.648e-02  8.857e-02   0.638 0.523673
instrate    -3.521e-01  9.981e-02  -3.527 0.000420 ***
malediv     -5.518e-01  4.449e-01  -1.240 0.214802
malesingle   5.595e-01  2.374e-01   2.357 0.018411 *
malemarwid   8.168e-02  3.421e-01   0.239 0.811311
coapp       -8.291e-01  5.111e-01  -1.622 0.104779
guar         7.445e-01  4.563e-01   1.631 0.102801
presres     -1.008e-01  9.679e-02  -1.042 0.297500
realest      1.915e-01  2.466e-01   0.776 0.437475
propnone    -1.797e-01  4.564e-01  -0.394 0.693857
age          1.827e-02  1.019e-02   1.792 0.073173 .
other       -3.604e-01  2.362e-01  -1.526 0.126976
rent        -2.345e-01  5.621e-01  -0.417 0.676537
ownres      -5.753e-02  5.371e-01  -0.107 0.914702
```

```
numcred     -2.412e-01  1.868e-01  -1.291 0.196741
job         -3.661e-02  1.648e-01  -0.222 0.824244
numdep      -3.114e-01  2.904e-01  -1.072 0.283523
tel          2.516e-01  2.268e-01   1.109 0.267366
for.         1.536e+00  8.062e-01   1.905 0.056781 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 916.30  on 749  degrees of freedom
Residual deviance: 689.55  on 719  degrees of freedom
AIC: 751.55

Number of Fisher Scoring iterations: 5


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

         0   1
  FALSE  40  18
  TRUE   35 157

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

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5475  -0.8448   0.4505   0.7671   1.8101

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
chkacct     5.977e-01  8.075e-02   7.402 1.34e-13 ***
dur        -1.820e-02  9.416e-03  -1.933 0.053291 .
hist        3.887e-01  8.303e-02   4.682 2.84e-06 ***
amt        -1.388e-04  4.374e-05  -3.175 0.001499 **
sav         2.184e-01  6.581e-02   3.318 0.000907 ***
```

```
instrate   -3.168e-01  8.001e-02  -3.959 7.52e-05 ***
malesingle  4.671e-01  1.927e-01   2.424 0.015344 *
age         1.318e-02  7.300e-03   1.805 0.071051 .
for.        1.420e+00  7.882e-01   1.802 0.071539 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1039.72  on 750  degrees of freedom
Residual deviance:  732.24  on 741  degrees of freedom
AIC: 750.24

Number of Fisher Scoring iterations: 5


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

         0   1
  FALSE  33  15
  TRUE   42 160

> #3m)
> library(ROCR)
> predrocr2 <- prediction(pred2, test$resp)
> auc2 <- performance(predrocr2, measure = "auc")@y.values
> auc2
[[1]]
[1] 0.8292571

> predrocr3 <- prediction(pred3, test$resp)
> auc3 <- performance(predrocr3, measure = "auc")@y.values
> auc3
[[1]]
[1] 0.782781


> #3o)
> sortpred2 <- sort(pred2,decreasing=TRUE)
> sortpred2
       898        353        160        757        507        770        291        315
0.99363502 0.99221990 0.99210197 0.99196237 0.98896806 0.98842799 0.98789680 0.98650792
```

284	860	34	349	407	607	215	520
0.98628934	0.98602025	0.98565742	0.98404603	0.98374330	0.98223620	0.98093134	0.97954119
749	428	761	895	717	235	782	150
0.97890890	0.97812096	0.97807812	0.97778435	0.97593435	0.97556472	0.97239924	0.97220592
174	86	934	818	909	914	750	491
0.97216123	0.97078777	0.96978698	0.96790341	0.96718244	0.96463297	0.96376241	0.96343739
961	942	711	366	892	985	551	743
0.96134356	0.95944148	0.95937129	0.95767579	0.95757508	0.95705820	0.95588025	0.95434725
97	326	259	904	451	449	412	899
0.95253502	0.95048209	0.95007593	0.94978236	0.94930182	0.94719349	0.94675604	0.94291044
373	842	413	995	151	255	573	270
0.94279648	0.94081009	0.94030819	0.94000036	0.93926207	0.93862183	0.93464807	0.93236727
103	774	448	905	385	454	902	372
0.93228280	0.93132748	0.93094780	0.92949963	0.92783878	0.92776617	0.92664323	0.92574501
675	699	118	958	299	438	837	194
0.92475666	0.92377398	0.92328348	0.92325110	0.92237876	0.91987242	0.91551197	0.91460674
739	890	294	57	307	881	560	862
0.91286659	0.91024441	0.90830959	0.90503903	0.90257163	0.90201756	0.90103571	0.89969124
846	76	188	609	599	345	887	839
0.89876257	0.89863025	0.89851725	0.89521729	0.89275002	0.89161501	0.88882844	0.88825641
660	718	331	495	585	47	425	172
0.88804192	0.88767028	0.88635441	0.88285536	0.88164880	0.88064729	0.87654976	0.87498556
622	503	306	648	390	996	534	450
0.87199126	0.86939059	0.86844495	0.86661634	0.86150266	0.85567792	0.85047487	0.84697945
910	95	572	796	937	710	13	834
0.84689454	0.84688313	0.84440932	0.84061731	0.83987435	0.83853433	0.83534618	0.83432521
527	844	251	771	701	409	731	43
0.83087169	0.82990819	0.82891058	0.82731455	0.82281261	0.82085014	0.81279612	0.80361324
612	583	851	968	278	22	46	561
0.80216976	0.79942662	0.79675070	0.79482240	0.79470155	0.79449965	0.78451396	0.77940734
458	500	336	73	305	952	591	40
0.77919223	0.77709249	0.77617417	0.76749179	0.76098168	0.76083007	0.75995800	0.75635448
822	780	751	992	111	876	344	94
0.75599299	0.75382829	0.75206866	0.75031298	0.74710291	0.74470190	0.73988687	0.72980994
972	704	588	824	566	965	829	982
0.72693598	0.72690714	0.72635036	0.72549426	0.72023045	0.71874947	0.71763286	0.71665295
580	293	906	911	87	119	504	575
0.71543181	0.70529318	0.70381381	0.70205710	0.69874148	0.67385701	0.67131107	0.67007847
617	605	181	690	433	435	931	14
0.66747141	0.66561230	0.66146680	0.65902569	0.65726275	0.65550038	0.65032374	0.64976505
324	430	592	420	700	760	231	949
0.64648816	0.64613399	0.64066111	0.63867871	0.63487595	0.63399751	0.62656756	0.62104712
636	127	38	288	462	691	530	159
0.60952732	0.60366765	0.60148179	0.59747992	0.58992538	0.58832457	0.58057621	0.57856241

```
       69        732        763        467        411        673        879        920 
0.56552206 0.53076594 0.52083757 0.51218111 0.50859050 0.50476689 0.50454342 0.50353702 
      114        508          4        488        482        532        826         16 
0.49709227 0.49510511 0.49027302 0.48821783 0.48669756 0.47999246 0.47704129 0.47611382 
      511        814        107        638        340        955        131        238 
0.47475141 0.47364939 0.46341751 0.45010258 0.44350939 0.44219569 0.44198077 0.44047537 
        2        863        900        546        158        915        241        606 
0.42640194 0.42389880 0.42364762 0.42248968 0.41777408 0.40773187 0.39652387 0.38031527 
      776        875        640        369        999        442        870        204 
0.37701296 0.37514784 0.36887384 0.36767112 0.36257907 0.35957941 0.35914779 0.35619745 
      287        756        209        841        897        886        653        641 
0.35322641 0.33717239 0.31621057 0.31238604 0.30926356 0.29401936 0.28749030 0.27758014 
      918        549        322        472        729        728        916        275 
0.27456413 0.27285052 0.23518835 0.22307414 0.22022625 0.19054183 0.18212131 0.17975788 
      432        740        132         64        854        836        171         60 
0.17815089 0.16122867 0.16090176 0.15155577 0.15104901 0.14774176 0.12110306 0.10727777 
      273        819 
0.06564072 0.02534326 
> germancredit[819,]
    chkacct dur hist newcar usedcar furn radiotv educ retrain   amt sav emp instrate
819       0  36    2      0       0    0       0    0       0 15857   0   0        2
    malediv malesingle malemarwid coapp guar presres realest propnone age other rent ownres
819       1          0          0     1    0       3       0        0  43     0    0      1
    numcred job numdep tel for. resp
819       1   3      1   0    0    1

> #3p)
> sortpred2 <- sort(pred2,decreasing=TRUE,index.return=TRUE)
> sortpred2$x
      898        353        160        757        507        770        291        315 
0.99363502 0.99221990 0.99210197 0.99196237 0.98896806 0.98842799 0.98789680 0.98650792 
      284        860         34        349        407        607        215        520 
0.98628934 0.98602025 0.98565742 0.98404603 0.98374330 0.98223620 0.98093134 0.97954119 
      749        428        761        895        717        235        782        150 
0.97890890 0.97812096 0.97807812 0.97778435 0.97593435 0.97556472 0.97239924 0.97220592 
      174         86        934        818        909        914        750        491 
0.97216123 0.97078777 0.96978698 0.96790341 0.96718244 0.96463297 0.96376241 0.96343739 
      961        942        711        366        892        985        551        743 
0.96134356 0.95944148 0.95937129 0.95767579 0.95757508 0.95705820 0.95588025 0.95434725 
       97        326        259        904        451        449        412        899 
0.95253502 0.95048209 0.95007593 0.94978236 0.94930182 0.94719349 0.94675604 0.94291044 
      373        842        413        995        151        255        573        270 
0.94279648 0.94081009 0.94030819 0.94000036 0.93926207 0.93862183 0.93464807 0.93236727 
      103        774        448        905        385        454        902        372 
```

0.93228280	0.93132748	0.93094780	0.92949963	0.92783878	0.92776617	0.92664323	0.92574501
675	699	118	958	299	438	837	194
0.92475666	0.92377398	0.92328348	0.92325110	0.92237876	0.91987242	0.91551197	0.91460674
739	890	294	57	307	881	560	862
0.91286659	0.91024441	0.90830959	0.90503903	0.90257163	0.90201756	0.90103571	0.89969124
846	76	188	609	599	345	887	839
0.89876257	0.89863025	0.89851725	0.89521729	0.89275002	0.89161501	0.88882844	0.88825641
660	718	331	495	585	47	425	172
0.88804192	0.88767028	0.88635441	0.88285536	0.88164880	0.88064729	0.87654976	0.87498556
622	503	306	648	390	996	534	450
0.87199126	0.86939059	0.86844495	0.86661634	0.86150266	0.85567792	0.85047487	0.84697945
910	95	572	796	937	710	13	834
0.84689454	0.84688313	0.84440932	0.84061731	0.83987435	0.83853433	0.83534618	0.83432521
527	844	251	771	701	409	731	43
0.83087169	0.82990819	0.82891058	0.82731455	0.82281261	0.82085014	0.81279612	0.80361324
612	583	851	968	278	22	46	561
0.80216976	0.79942662	0.79675070	0.79482240	0.79470155	0.79449965	0.78451396	0.77940734
458	500	336	73	305	952	591	40
0.77919223	0.77709249	0.77617417	0.76749179	0.76098168	0.76083007	0.75995800	0.75635448
822	780	751	992	111	876	344	94
0.75599299	0.75382829	0.75206866	0.75031298	0.74710291	0.74470190	0.73988687	0.72980994
972	704	588	824	566	965	829	982
0.72693598	0.72690714	0.72635036	0.72549426	0.72023045	0.71874947	0.71763286	0.71665295
580	293	906	911	87	119	504	575
0.71543181	0.70529318	0.70381381	0.70205710	0.69874148	0.67385701	0.67131107	0.67007847
617	605	181	690	433	435	931	14
0.66747141	0.66561230	0.66146680	0.65902569	0.65726275	0.65550038	0.65032374	0.64976505
324	430	592	420	700	760	231	949
0.64648816	0.64613399	0.64066111	0.63867871	0.63487595	0.63399751	0.62656756	0.62104712
636	127	38	288	462	691	530	159
0.60952732	0.60366765	0.60148179	0.59747992	0.58992538	0.58832457	0.58057621	0.57856241
69	732	763	467	411	673	879	920
0.56552206	0.53076594	0.52083757	0.51218111	0.50859050	0.50476689	0.50454342	0.50353702
114	508	4	488	482	532	826	16
0.49709227	0.49510511	0.49027302	0.48821783	0.48669756	0.47999246	0.47704129	0.47611382
511	814	107	638	340	955	131	238
0.47475141	0.47364939	0.46341751	0.45010258	0.44350939	0.44219569	0.44198077	0.44047537
2	863	900	546	158	915	241	606
0.42640194	0.42389880	0.42364762	0.42248968	0.41777408	0.40773187	0.39652387	0.38031527
776	875	640	369	999	442	870	204
0.37701296	0.37514784	0.36887384	0.36767112	0.36257907	0.35957941	0.35914779	0.35619745
287	756	209	841	897	886	653	641
0.35322641	0.33717239	0.31621057	0.31238604	0.30926356	0.29401936	0.28749030	0.27758014
918	549	322	472	729	728	916	275

```
0.27456413 0.27285052 0.23518835 0.22307414 0.22022625 0.19054183 0.18212131 0.17975788
       432        740        132         64        854        836        171         60
0.17815089 0.16122867 0.16090176 0.15155577 0.15104901 0.14774176 0.12110306 0.10727777
       273        819
0.06564072 0.02534326
> sortpred2$ix
  [1] 218  78  37 176 115 180  61  68  58 204   7  77  85 141  46 118 172  92 178 216 163
 [22]  48 185  33  40  19 234 188 225 228 173 110 241 236 162  79 215 246 125 171  23  71
 [43]  53 222 102 100  88 219  82 199  89 248  34  52 130  54  24 182  99 223  83 103 221
 [64]  81 154 157  28 240  64  97 196  43 169 214  63  13  67 211 126 205 201  18  42 142
 [85] 138  76 213 197 152 164  72 111 134  12  91  39 145 113  66 150  84 249 122 101 226
[106]  22 129 186 235 161   3 194 119 200  51 181 159  86 167  10 143 133 202 243  57   6
[127]  11 127 104 112  73  17  65 238 136   9 190 184 174 247  26 209  75  21 244 160 135
[148] 191 128 242 193 245 132  62 224 227  20  29 114 131 144 139  41 155  95  96 233   4
[169]  70  93 137  90 158 177  47 237 146  30   8  60 105 156 120  36  16 168 179 106  87
[190] 153 210 232  27 116   2 109 108 121 192   5 117 187  25 147  74 239  31  49   1 206
[211] 220 123  35 229  50 140 183 208 148  80 250  98 207  44  59 175  45 198 217 212 151
[232] 149 231 124  69 107 166 165 230  56  94 170  32  15 203 195  38  14  55 189
> test$resp[sortpred2$ix]
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [44] 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 0 1
 [87] 1 1 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 1 0
[130] 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 0 1 0 1 0
[173] 1 0 0 0 1 1 0 1 1 1 1 1 0 0 1 0 1 1 0 0 0 0 1 1 1 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 0
[216] 1 0 1 0 0 0 1 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
> profitpred2 <- 100*test$resp[sortpred2$ix] -300*(1-test$resp[sortpred2$ix])
> cumulative <- cumsum(profitpred2)
> max(cumulative)
[1] 7800
> which.max(cumulative)
[1] 150

> #3q)
> sortpred2$x[which.max(cumulative)]
      965
0.7187495
```

4. There has been significant interest in the results of the U.S. elections in November 2016. In this question you will build a model that would have helped predict the winner of the U.S. presidential elections using data which was available before the elections. The model will use past election outcomes and the state of the economy to predict how people might vote. The data is provided in the file **presidential.csv** and contains the following variables:

   - **YEAR**: Year of the U.S. presidential election
   - **DEM**: Name of Democratic nominee
   - **REP**: Name of Republican nominee
   - **INC**: Incumbent (party in power) leading up to that election (1 = Democratic, -1 = Republican)
   - **RUN**: Variable to indicate if the incumbent president is running for the presidential election again (1 = Democratic incumbent president is running, -1 = Republican incumbent president is running, 0 = otherwise. For example after becoming president in 2008, Obama ran for presidential elections again as a Democrat in 2012 implying that the corresponding entry is 1.)
   - **DUR**: Duration of the current party in power in the White House (0 = Incumbent has been in power only for one term before the election, 1 (-1) if the Democratic (Republican) party has been in the White House for two consecutive terms, 1.25 (-1.25) if the Democratic (Republican) party has been in the White House for three consecutive terms, 1.50 (-1.50) if the Democratic (Republican) party has been in the White House for four consecutive terms, and so on)
   - **GROWTH**: Growth rate of the real per capita GDP in the year of the election (%)
   - **GOOD**: Number of good quarters (in terms of performance in the growth rate of the real capita per GDP) in the first fifteen quarters of the current administration
   - **WIN**: Winner of the presidential election (1 = Democratic, -1 = Republican)

   (a) Read the dataset into the dataframe **pres**. In the elections starting from 1916 up to and including 2012, which party has won more presidential elections? How many elections has that party won?

   Solution. There have been 14 Democrat president winners from 1916 to 2012 and 11 Republican winners.

   (b) Who among the nominees have represented the Democrats and the Republicans in the most number of presidential elections? How many times have they respectively done so?

   Solution. Roosevelt has represented Democrats 4 times in elections, and Nixon has represented Republicans 3 times in elections.

(c) Use a two-sided t-test to verify if there is evidence to show that the number of good quarters when the president is Republican is different from the number of good quarters when the president is Democratic. What is the p-value of the test and your conclusion?

Solution. The p-value for the two-sided t-test is 0.7494, which implies 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) Define a new variable **WININC** that takes a value of 1 if the presidential nominee of the incumbent party wins and 0 otherwise. Provide the R command(s) that you used to create this variable.

Solution. `pres$WININC <- as.integer(pres$INC == pres$WIN)`

(e) How many times did the presidential nominee from the incumbent party win and how many times did the presidential nominee from the incumbent party lose?

Solution. The incumbent won 16 times, and lost 9.

(f) Perform a simple logistic regression to predict the **WININC** variable using the **GROWTH** variable and the constant. What is the log-likelihood value for the model?

Solution.

$$LL(\hat{\beta}) = \frac{2p - AIC}{2} = \frac{2(2) - 30.365}{2} = -13.1825.$$

(g) The **GROWTH** variable is:

- Significant at the 0.001 level
- Significant at the 0.01 level
- Significant at the 0.05 level
- Significant at the 0.1 level
- Insignificant

Solution. Since the p-value for the $GROWTH$ variable under null hypothesis $H_0 : \beta_1 = 0$ is 0.0613, it si significant at the 0.1 level.

(h) Unlike questions (d) to (g) which looked at the incumbent party's winning chances, from this point onwards, we are going to predict the chances of the Democratic party nominee winning in the presidential election. To do this, we need to transform the variables as follows:

i. Transform the **WIN** variable to be 1 when the presidential winner is a Democrat and 0 when the winner is a Republican.

ii. Transform the **GROWTH** variable as follows: When the growth rate is positive (say 4.623) and the Republican party is incumbent, we should transform it to a negative value -4.623 since this should have a negative effect on the Democratic nominee's chances of winning while if the growth rate is negative (say -4.623) and the Republican party is incumbent, we should transform it to positive 4.623 since this should have a positive effect on the Democratic nominee's chances of winning.

Write down the R command(s) for i and ii.

Solution. `pres$WIN <- as.integer(pres$WIN == 1)`

`pres$GROWTH <- pres$GROWTH * pres$INC`

(i) Repeat step ii in question (h) for the **GOOD** variable. You are now ready to develop a logistic regression model for the **WIN** variable using the predictor variables **INC**, **RUN**, **DUR**, **GROWTH**, **GOOD** and the constant (intercept). Use all the observations to build your model. What is the AIC of the model?

Solution. AIC of the model is 29.406.

(j) Among the predictor variables **INC**, **RUN**, **DUR**, **GROWTH**, **GOOD** and the constant (intercept), identify the three least significant variables?

Solution. The least significant variables in the model are: *Intercept* (0.941), *INC* (0.955) and *GOOD* (0.728).

(k) Drop the three variables identified in question (j) and rebuild you logistic regression model. What is the AIC of the new model?

Solution. AIC for the new model is 23.748.

(l) In this new model, what is the smallest significance level at which you would reject the null hypothesis that the coefficient for the variable **DUR** is zero? Suppose, we now decide to use a level of 0.10, what would your suggestion be?

Solution. 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, we might be inclined to leave it in the model anyway.

(m) Which among the two models that you have developed in questions (i) and (k) do you prefer? Explain your reasons briefly.

Solution. The second model (*RUN, GROWTH, DUR*) has a lower AIC value. By dropping the variables deemed insignificant to form this model, we also have a more interpretable model.

(n) We will now evaluate the probability of H.Clinton winning the 2016 election with this model where H.Clinton is the Democratic nominee and D.Trump is the Republican nominee. What should be the corresponding **INC**, **RUN** and **DUR** variables?

Solution. We have $INC = 1$, since the Democrats are in power. $RUN = 0$, since Obama did not run, and $DUR = 1$, since the Democrats have been in power for 2 consecutive terms.

(o) The forecasted growth rate from analysts of the U.S. economy for this year is 2%. Based on this, what is the probability of H.Clinton winning in the upcoming election based on the model you developed in question (11)?

Solution. $GROWTH = 2$. We have:

$$P(Dem = 1) = \frac{e^{2.0638(0)+0.4690(2)-1.7852(1)}}{1 + e^{2.0638(0)+0.4690(2)-1.7852(1)}} = 0.3.$$

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

R Scripts.

```
> #4a)
> pres <- read.csv("presidential.csv")
> str(pres)
'data.frame': 25 obs. of  9 variables:
 $ YEAR  : int  1916 1920 1924 1928 1932 1936 1940 1944 1948 1952 ...
 $ DEM   : Factor w/ 18 levels "Carter","Clinton",..: 18 3 4 15 14 14 14 14 17 16 ...
 $ REP   : Factor w/ 17 levels "Coolidge","Dewey",..: 11 9 1 10 10 12 17 2 2 4 ...
 $ INC   : int  1 1 -1 -1 -1 1 1 1 1 1 ...
 $ RUN   : int  1 0 -1 0 -1 1 1 1 1 0 ...
 $ DUR   : num  0 1 0 -1 -1.25 0 1 1.25 1.5 1.75 ...
 $ GROWTH: num  2.23 -11.46 -3.87 4.62 -14.35 ...
 $ GOOD  : int  3 0 10 7 4 9 8 0 0 7 ...
 $ WIN   : int  1 -1 -1 -1 1 1 1 1 1 -1 ...
> table(pres$WIN)

-1  1
11 14
```

```
> #4b)
> sort(table(pres$DEM))

      Cox      Davis    Dukakis       Gore   Humphrey    Johnson    Kennedy      Kerry   McGovern
        1          1          1          1          1          1          1          1          1
  Mondale      Smith     Truman     Wilson     Carter    Clinton      Obama  Stevenson  Roosevelt
        1          1          1          1          2          2          2          2          4
> sort(table(pres$REP))

  Coolidge       Dole       Ford  Goldwater    Harding     Hughes     Landon     McCain
         1          1          1          1          1          1          1          1
    Romney     Wilkie      Dewey Eisenhower     G.Bush   G.W.Bush     Hoover     Reagan
         1          1          2          2          2          2          2          2
     Nixon
         3

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

Welch Two Sample t-test

data:  pres$GOOD[pres$INC == 1] and pres$GOOD[pres$INC == -1]
t = -0.32362, df = 21.196, p-value = 0.7494
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.854816  2.085585
sample estimates:
mean of x mean of y
 4.615385  5.000000


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

> #4e)
> table(pres$WININC)

 0  1
 9 16

> #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)

Call:
glm(formula = WIN ~ INC + RUN + GROWTH + DUR + GOOD, family = binomial,
    data = pres)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6418  -0.3862   0.0302   0.3889   1.6105

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.04699    0.63096   0.074    0.941
INC         -0.17339    3.06255  -0.057    0.955
RUN          1.96642    1.71162   1.149    0.251
GROWTH       0.50456    0.31041   1.625    0.104
DUR         -2.08399    1.77249  -1.176    0.240
GOOD         0.10169    0.29263   0.348    0.728

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.296  on 24  degrees of freedom
Residual deviance: 17.406  on 19  degrees of freedom
AIC: 29.406

Number of Fisher Scoring iterations: 7


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

Call:
glm(formula = WIN ~ RUN + GROWTH + DUR - 1, family = binomial,
    data = pres)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
```

```
-1.46675  -0.36836   0.04492   0.47750   1.73580

Coefficients:
       Estimate Std. Error z value Pr(>|z|)
RUN      2.0638     0.9772   2.112   0.0347 *
GROWTH   0.4690     0.2774   1.691   0.0909 .
DUR     -1.7852     1.0876  -1.641   0.1007
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.657  on 25  degrees of freedom
Residual deviance: 17.748  on 22  degrees of freedom
AIC: 23.748

Number of Fisher Scoring iterations: 6
```