Lab 11: Variable Selection and Logistic Regression

Stat 131A (Fall 2021)

Due: Tuesday November 30, 2021

Welcome to the Lab 11! In the first part, we will apply variable selection techniques to find the best subset of covariates to predict the red wine quality using physicochemical tests scores such as citric acid, pH, etc.

The dataset we will be using is related to red variants of the Portuguese *vinho verde* wine. There are 1599 samples available in the dataset. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

The explanatory variables are all continuous variables and based on physicochemical tests:

- **fixed acidity**
- **volatile acidity**
- **citric acid**
- **residual sugar**
- **chlorides**
- **free sulfur dioxide**
- **total sulfur dioxide**
- **density**
- **pH**
- **sulphates**
- **alcohol**

The response variable is the **quality** score between 0 and 10 (based on sensory data).

We randomly split the data into two parts-the `wine` dataset with 1199 samples and the `wine.test` dataset with 400 samples. Splitting the dataset is a common technique when we want to evaluate the model performance. There are training set, validation set, and test set. The validation set is used for model selection. That is, to estimate the performance of the different model in order to choose the best one. The test set is used for estimating the performance of our final model.

```
set.seed(20170413)
wine.dataset <- read.csv("winequality-red.csv", sep = ";")
test.samples <- sample(1:nrow(wine.dataset), 400)
wine <- wine.dataset[-test.samples, ]
wine.test <- wine.dataset[test.samples, ]
```

We now fit a linear regression using all of the explanatory variables:

```
wine.fit <- lm(quality ~. ,data = na.omit(wine))
summary(wine.fit)
```

```
##
## Call:
## lm(formula = quality ~ ., data = na.omit(wine))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.69755 -0.35429 -0.03872  0.42375  1.99847
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.093e+01  2.416e+01   0.867   0.3864
## fixed.acidity         3.130e-02  2.980e-02   1.050   0.2938
## volatile.acidity     -1.163e+00  1.373e-01  -8.465  < 2e-16 ***
## citric.acid          -3.944e-01  1.649e-01  -2.392   0.0169 *
## residual.sugar        2.289e-02  1.693e-02   1.351   0.1768
## chlorides            -2.191e+00  4.821e-01  -4.544 6.09e-06 ***
## free.sulfur.dioxide   6.202e-03  2.407e-03   2.576   0.0101 *
## total.sulfur.dioxide -3.471e-03  8.193e-04  -4.237 2.44e-05 ***
## density              -1.699e+01  2.468e+01  -0.688   0.4913
## pH                   -4.129e-01  2.176e-01  -1.898   0.0580 .
## sulphates             1.022e+00  1.311e-01   7.802 1.33e-14 ***
## alcohol               2.860e-01  2.991e-02   9.562  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6314 on 1187 degrees of freedom
## Multiple R-squared:  0.3891, Adjusted R-squared:  0.3834
## F-statistic: 68.72 on 11 and 1187 DF,  p-value: < 2.2e-16
```

Exercise 1: Backward elimination based on p-values

We start with our full model `wine.fit`.

(a) Remove the term corresponding to the coefficient estimate with the highest p-value in the full model. Print the summary of your updated model.

```
# Insert your code here and save your updated model as `wine.backward`
wine.backward <- lm(quality ~ .-density,data = na.omit(wine))
summary(wine.backward)
```

```
##
## Call:
## lm(formula = quality ~ . - density, data = na.omit(wine))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.68059 -0.35615 -0.03571  0.42659  2.01281
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           4.3100882  0.6873903   6.270 5.04e-10 ***
## fixed.acidity         0.0149461  0.0179866   0.831  0.40617
## volatile.acidity     -1.1749858  0.1361037  -8.633  < 2e-16 ***
## citric.acid          -0.3944298  0.1648857  -2.392  0.01691 *
## residual.sugar        0.0156751  0.0133024   1.178  0.23889
## chlorides            -2.2216268  0.4799194  -4.629 4.07e-06 ***
## free.sulfur.dioxide   0.0063582  0.0023963   2.653  0.00808 **
## total.sulfur.dioxide -0.0035260  0.0008152  -4.325 1.65e-05 ***
## pH                   -0.5014344  0.1754039  -2.859  0.00433 **
## sulphates             0.9986542  0.1263766   7.902 6.21e-15 ***
## alcohol               0.3015574  0.0196246  15.366  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6313 on 1188 degrees of freedom
## Multiple R-squared:  0.3888, Adjusted R-squared:  0.3837
## F-statistic: 75.58 on 10 and 1188 DF,  p-value: < 2.2e-16
```

(b) In R, there are functions which automatically perform variable selection. The `step()` function uses AIC, which is very similar to RSS but also takes the number of explanatory variables into account. For example, to do backward elimination starting with our full model:

```
step(wine.fit, direction = "backward")
```

```
## Start:  AIC=-1090.65
## quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     density + pH + sulphates + alcohol
##
##                       Df Sum of Sq    RSS     AIC
## - density              1     0.189 473.43 -1092.2
## - fixed.acidity        1     0.440 473.68 -1091.5
## - residual.sugar       1     0.728 473.97 -1090.8
## <none>                             473.24 -1090.7
## - pH                   1     1.436 474.67 -1089.0
## - citric.acid          1     2.281 475.52 -1086.9
## - free.sulfur.dioxide  1     2.646 475.88 -1086.0
## - total.sulfur.dioxide 1     7.156 480.39 -1074.7
## - chlorides            1     8.231 481.47 -1072.0
## - sulphates            1    24.266 497.50 -1032.7
## - volatile.acidity     1    28.567 501.80 -1022.4
## - alcohol              1    36.456 509.69 -1003.7
##
## Step:  AIC=-1092.17
## quality ~ fixed.acidity + volatile.acidity + citric.acid + residual.sugar +
##     chlorides + free.sulfur.dioxide + total.sulfur.dioxide +
##     pH + sulphates + alcohol
##
##                       Df Sum of Sq    RSS      AIC
## - fixed.acidity        1     0.275 473.70 -1093.47
## - residual.sugar       1     0.553 473.98 -1092.77
## <none>                             473.43 -1092.17
## - citric.acid          1     2.280 475.71 -1088.41
## - free.sulfur.dioxide  1     2.806 476.23 -1087.08
## - pH                   1     3.257 476.68 -1085.95
## - total.sulfur.dioxide 1     7.456 480.88 -1075.43
## - chlorides            1     8.540 481.97 -1072.73
## - sulphates            1    24.885 498.31 -1032.74
## - volatile.acidity     1    29.700 503.13 -1021.21
## - alcohol              1    94.097 567.52  -876.81
##
## Step:  AIC=-1093.47
## quality ~ volatile.acidity + citric.acid + residual.sugar + chlorides +
##     free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates +
##     alcohol
##
```

```
##                         Df Sum of Sq    RSS      AIC
## - residual.sugar         1     0.590 474.29 -1093.98
## <none>                               473.70 -1093.47
## - citric.acid            1     2.140 475.84 -1090.07
## - free.sulfur.dioxide    1     2.940 476.64 -1088.05
## - pH                     1     5.910 479.61 -1080.60
## - total.sulfur.dioxide   1     8.930 482.63 -1073.08
## - chlorides              1     9.930 483.63 -1070.60
## - sulphates              1    25.248 498.95 -1033.21
## - volatile.acidity       1    30.044 503.75 -1021.74
## - alcohol                1    94.495 568.20  -877.39
##
## Step:  AIC=-1093.98
## quality ~ volatile.acidity + citric.acid + chlorides + free.sulfur.dioxide +
##     total.sulfur.dioxide + pH + sulphates + alcohol
##
##                         Df Sum of Sq    RSS      AIC
## <none>                               474.29 -1093.98
## - citric.acid            1     1.867 476.16 -1091.27
## - free.sulfur.dioxide    1     3.331 477.62 -1087.59
## - pH                     1     5.975 480.27 -1080.97
## - total.sulfur.dioxide   1     8.638 482.93 -1074.34
## - chlorides              1     9.691 483.98 -1071.73
## - sulphates              1    24.867 499.16 -1034.71
## - volatile.acidity       1    29.500 503.79 -1023.63
## - alcohol                1    96.007 570.30  -874.96
##
## Call:
## lm(formula = quality ~ volatile.acidity + citric.acid + chlorides +
##     free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates +
##     alcohol, data = na.omit(wine))
##
## Coefficients:
##          (Intercept)      volatile.acidity           citric.acid
##             4.697601             -1.131930              -0.294894
##            chlorides   free.sulfur.dioxide  total.sulfur.dioxide
##            -2.288882              0.006858              -0.003640
##                   pH             sulphates               alcohol
##            -0.580238              0.994885               0.300961
```

Now try to understand the output of `step()` function. Which variables were omitted from the final model? Provide a list of those variables in order of their elimination, and write the final model.

Variables eliminated (in order):density-fixed.acidity-residual.sugar

Final model: quality ~ volatile.acidity + citric.acid + chlorides + free.sulfur.dioxide + total.sulfur.dioxide + pH + sulphates + alcohol

(c) Start from the model with only intercept term. Use the `step()` function to perform forward selection. Write the variables added in order of their addition and the final model.
Hint. (a) Use the `scope` argument in step function. (b) Use `formula()` function to get the formula of your full model.

```
# Insert your code here
wine.start = lm(quality ~ 1 ,data = na.omit(wine))
```

```r
wine.start.fwd = step(
  wine.start,
  scope = formula(wine.fit),
  direction = "forward")
```

```
## Start:  AIC=-521.79
## quality ~ 1
##
##                        Df Sum of Sq    RSS     AIC
## + alcohol               1   192.427 582.20 -862.19
## + volatile.acidity      1   119.348 655.28 -720.41
## + sulphates             1    56.804 717.82 -611.11
## + citric.acid           1    30.890 743.74 -568.59
## + density               1    29.454 745.17 -566.27
## + total.sulfur.dioxide  1    28.192 746.44 -564.24
## + chlorides             1    15.129 759.50 -543.44
## + fixed.acidity         1     8.531 766.10 -533.07
## + pH                    1     1.596 773.03 -522.27
## <none>                              774.63 -521.79
## + free.sulfur.dioxide   1     0.595 774.03 -520.72
## + residual.sugar        1     0.026 774.60 -519.83
##
## Step:  AIC=-862.19
## quality ~ alcohol
##
##                        Df Sum of Sq    RSS      AIC
## + volatile.acidity      1    67.673 514.53 -1008.34
## + sulphates             1    35.602 546.60  -935.85
## + pH                    1    16.389 565.81  -894.42
## + citric.acid           1    15.880 566.32  -893.35
## + fixed.acidity         1    14.456 567.75  -890.33
## + total.sulfur.dioxide  1     5.078 577.12  -870.69
## + density               1     2.941 579.26  -866.26
## + chlorides             1     0.983 581.22  -862.21
## <none>                              582.20  -862.19
## + free.sulfur.dioxide   1     0.120 582.08  -860.44
## + residual.sugar        1     0.000 582.20  -860.19
##
## Step:  AIC=-1008.34
## quality ~ alcohol + volatile.acidity
##
##                        Df Sum of Sq    RSS     AIC
## + sulphates             1   16.2128 498.32 -1044.7
## + total.sulfur.dioxide  1    4.0390 510.49 -1015.8
## + pH                    1    3.0633 511.47 -1013.5
## + fixed.acidity         1    2.6034 511.93 -1012.4
## + density               1    1.0256 513.50 -1008.7
## + chlorides             1    0.9571 513.57 -1008.6
## <none>                              514.53 -1008.3
## + citric.acid           1    0.3324 514.20 -1007.1
## + residual.sugar        1    0.0264 514.50 -1006.4
## + free.sulfur.dioxide   1    0.0097 514.52 -1006.4
##
## Step:  AIC=-1044.73
```

```
## quality ~ alcohol + volatile.acidity + sulphates
##
##                        Df Sum of Sq    RSS     AIC
## + chlorides             1    9.0441 489.27 -1064.7
## + total.sulfur.dioxide  1    5.5025 492.81 -1056.0
## + citric.acid           1    2.1128 496.20 -1047.8
## + fixed.acidity         1    1.2818 497.03 -1045.8
## + pH                    1    1.2187 497.10 -1045.7
## <none>                              498.32 -1044.7
## + free.sulfur.dioxide   1    0.0388 498.28 -1042.8
## + density               1    0.0282 498.29 -1042.8
## + residual.sugar        1    0.0076 498.31 -1042.8
##
## Step:  AIC=-1064.69
## quality ~ alcohol + volatile.acidity + sulphates + chlorides
##
##                        Df Sum of Sq    RSS     AIC
## + total.sulfur.dioxide  1    6.0221 483.25 -1077.5
## + pH                    1    2.9442 486.33 -1069.9
## + fixed.acidity         1    1.6005 487.67 -1066.6
## <none>                              489.27 -1064.7
## + citric.acid           1    0.5952 488.68 -1064.2
## + residual.sugar        1    0.1247 489.15 -1063.0
## + free.sulfur.dioxide   1    0.1005 489.17 -1062.9
## + density               1    0.0728 489.20 -1062.9
##
## Step:  AIC=-1077.54
## quality ~ alcohol + volatile.acidity + sulphates + chlorides +
##     total.sulfur.dioxide
##
##                       Df Sum of Sq    RSS     AIC
## + pH                   1   2.98298 480.27 -1083.0
## + free.sulfur.dioxide  1   2.92150 480.33 -1082.8
## + residual.sugar       1   0.82248 482.43 -1077.6
## + fixed.acidity        1   0.80760 482.44 -1077.5
## <none>                             483.25 -1077.5
## + citric.acid          1   0.33782 482.91 -1076.4
## + density              1   0.02353 483.23 -1075.6
##
## Step:  AIC=-1082.97
## quality ~ alcohol + volatile.acidity + sulphates + chlorides +
##     total.sulfur.dioxide + pH
##
##                       Df Sum of Sq    RSS     AIC
## + free.sulfur.dioxide  1    4.1082 476.16 -1091.3
## + citric.acid          1    2.6444 477.62 -1087.6
## <none>                             480.27 -1083.0
## + residual.sugar       1    0.5999 479.67 -1082.5
## + fixed.acidity        1    0.1059 480.16 -1081.2
## + density              1    0.0623 480.20 -1081.1
##
## Step:  AIC=-1091.27
## quality ~ alcohol + volatile.acidity + sulphates + chlorides +
##     total.sulfur.dioxide + pH + free.sulfur.dioxide
```

```
##
##                   Df Sum of Sq    RSS     AIC
## + citric.acid      1   1.86729 474.29 -1094.0
## <none>                         476.16 -1091.3
## + residual.sugar   1   0.31731 475.84 -1090.1
## + fixed.acidity    1   0.08918 476.07 -1089.5
## + density          1   0.04765 476.11 -1089.4
##
## Step:  AIC=-1093.98
## quality ~ alcohol + volatile.acidity + sulphates + chlorides +
##     total.sulfur.dioxide + pH + free.sulfur.dioxide + citric.acid
##
##                   Df Sum of Sq    RSS     AIC
## <none>                         474.29 -1094.0
## + residual.sugar   1   0.59005 473.70 -1093.5
## + fixed.acidity    1   0.31187 473.98 -1092.8
## + density          1   0.23861 474.05 -1092.6
```

Variables added (in order):alcohol + volatile.acidity + sulphates + chlorides + total.sulfur.dioxide + pH + free.sulfur.dioxide + citric.acid

Final model:quality ~ alcohol + volatile.acidity + sulphates + chlorides + total.sulfur.dioxide + pH + free.sulfur.dioxide + citric.acid

Exercise 2: Regression on all subsets of variables

To find the optimal subset of a certain number of variables for a regression, and to compare between different numbers of variables, use the `regsubsets()` function in the `leaps` package.

```
require(leaps)
```

```
## Loading required package: leaps
```

```
## Warning: package 'leaps' was built under R version 4.1.2
```

```
regsub_out <- regsubsets(x = wine[, -12] , y = wine[, 12])
summary(regsub_out)
```

```
## Subset selection object
## 11 Variables  (and intercept)
##                       Forced in Forced out
## fixed.acidity             FALSE      FALSE
## volatile.acidity          FALSE      FALSE
## citric.acid               FALSE      FALSE
## residual.sugar            FALSE      FALSE
## chlorides                 FALSE      FALSE
## free.sulfur.dioxide       FALSE      FALSE
## total.sulfur.dioxide      FALSE      FALSE
## density                   FALSE      FALSE
## pH                        FALSE      FALSE
## sulphates                 FALSE      FALSE
## alcohol                   FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1  ( 1 ) " "           " "              " "         " "            " "
## 2  ( 1 ) " "           "*"              " "         " "            " "
```

```
## 3  ( 1 ) " "           "*"              " "          " "            " "
## 4  ( 1 ) " "           "*"              " "          " "            "*"
## 5  ( 1 ) " "           "*"              " "          " "            "*"
## 6  ( 1 ) " "           "*"              " "          " "            "*"
## 7  ( 1 ) " "           "*"              " "          " "            "*"
## 8  ( 1 ) " "           "*"              "*"          " "            "*"
##          free.sulfur.dioxide total.sulfur.dioxide density pH  sulphates alcohol
## 1  ( 1 ) " "                 " "                  " "     " " " " "       "*"
## 2  ( 1 ) " "                 " "                  " "     " " " " "       "*"
## 3  ( 1 ) " "                 " "                  " "     " " "*"         "*"
## 4  ( 1 ) " "                 " "                  " "     " " "*"         "*"
## 5  ( 1 ) " "                 "*"                  " "     " " "*"         "*"
## 6  ( 1 ) " "                 "*"                  " "     "*" "*"         "*"
## 7  ( 1 ) "*"                 "*"                  " "     "*" "*"         "*"
## 8  ( 1 ) "*"                 "*"                  " "     "*" "*"         "*"
```

The default maximum subset size is `nvmax = 8`.

```
coef(regsub_out, 7)
```

```
##          (Intercept)     volatile.acidity            chlorides
##          4.152458457         -0.992176453        -2.456920286
##  free.sulfur.dioxide total.sulfur.dioxide                   pH
##          0.007546816         -0.003900354        -0.428126506
##            sulphates              alcohol
##          0.975879324          0.292841170
```

Optimal subsets of each size are chosen by RSS. To compare models with different subset sizes, use AIC.

```
coef(regsub_out, 1:3)
```

```
## [[1]]
## (Intercept)     alcohol
##   1.7135526   0.3758375
##
## [[2]]
##      (Intercept) volatile.acidity          alcohol
##        2.9291785       -1.3732552        0.3285815
##
## [[3]]
##      (Intercept) volatile.acidity        sulphates          alcohol
##        2.4393960       -1.1985154        0.7178874        0.3214194
```

What is the best model using 1 variable? Using 7? Is the optimal model of 7 covariates the same as that found in exercise 1?

Answer:
the best model using 1 variable is quality~alcohol
the best model using 7 variable is quality~volatile.acidity+chlorides+free.sulfur.dioxide+ total.sulfur.dioxide +pH+sulphates+alcohol
It's the same with the model found by forward selection in exercise 1.

Exercise 3: Compare performance using test set

Use the test set to assess the performance of the models resulting from forward stepwise selection and the full model. What is the test set root mean square error for the two models?

```
wine.forward <- lm(quality ~ alcohol + volatile.acidity + sulphates +
    chlorides + total.sulfur.dioxide + pH + free.sulfur.dioxide +
    citric.acid ,data = na.omit(wine.test))
RMSE_forward <- sqrt(mean(residuals(wine.forward)^2))
wine.full <- lm(quality ~. ,data = na.omit(wine.test))
RMSE_full <- sqrt(mean(residuals(wine.fit)^2))
RMSE_forward
```

```
## [1] 0.6799145
```

```
RMSE_full
```

```
## [1] 0.6282463
```

Answer here:
RMSE of full model is 0.6793743, RMSE of forward stepwise selection is 0.6282463

Logistic regression: customer retention

A telecommunications company is concerned about the number of customers leaving their landline business for cable competitors. They need to understand who is leaving. Imagine that you're an analyst at this company and you have to find out who is leaving and why.

We will use data from IBM Watson Analytics to predict customer retention. Analysis of relevant customer data can lead to the design of focused customer retention programs.

The data includes:

- Customers who left within the last month (column `Churn`)
- Services that each customer has signed up for – phone, multiple lines, internet, online security, online backup, device protection, tech support, and streaming TV and movies
- Customer account information – how long they've been a customer, contract, payment method, paperless billing, monthly charges, and total charges
- Demographic info about customers – gender, age range, and if they have partners and dependents

```
set.seed(131)
retention <- read.csv("customer_retention.csv", stringsAsFactors = FALSE)
retention$SeniorCitizen <- factor(retention$SeniorCitizen, 0:1, c("No", "Yes"))
retention <- retention[with(
    retention, MultipleLines != "No phone service" &
        OnlineSecurity != "No internet service"), ]
retention$Churn <- as.numeric(factor(retention$Churn, c("No", "Yes"))) - 1
retention$PhoneService = NULL
retention$PaymentMethod = factor(retention$PaymentMethod)
retention <- retention[, -which(names(retention) %in% c("customerID", "PhoneService"))]

test.set <- sample(nrow(retention), 500)
retention.test <- retention[test.set, ]
retention <- retention[-test.set, ]
head(retention)
```

```
##    gender SeniorCitizen Partner Dependents tenure MultipleLines InternetService
## 2    Male            No      No         No     34            No             DSL
## 5  Female            No      No         No      2            No     Fiber optic
## 6  Female            No      No         No      8           Yes     Fiber optic
## 7    Male            No      No        Yes     22           Yes     Fiber optic
## 9  Female            No     Yes         No     28           Yes     Fiber optic
```

```
## 11   Male            No     Yes        Yes     13            No             DSL
##    OnlineSecurity OnlineBackup DeviceProtection TechSupport StreamingTV
## 2             Yes           No               Yes          No          No
## 5              No           No                No          No          No
## 6              No           No               Yes          No         Yes
## 7              No          Yes                No          No         Yes
## 9              No           No               Yes         Yes         Yes
## 11            Yes           No                No          No          No
##    StreamingMovies       Contract PaperlessBilling           PaymentMethod
## 2               No       One year                No            Mailed check
## 5               No Month-to-month               Yes        Electronic check
## 6              Yes Month-to-month               Yes        Electronic check
## 7               No Month-to-month               Yes Credit card (automatic)
## 9              Yes Month-to-month               Yes        Electronic check
## 11              No Month-to-month               Yes            Mailed check
##    MonthlyCharges TotalCharges Churn
## 2           56.95      1889.50     0
## 5           70.70       151.65     1
## 6           99.65       820.50     1
## 7           89.10      1949.40     0
## 9          104.80      3046.05     1
## 11          49.95       587.45     0
```

Exercise 4

(a) Fit a logistic regression for `Churn` given all other variables in the dataset.

```
# insert your code here to fit a logistic regression.
glmRetention = glm(Churn ~ ., family = binomial, data = retention)
summary(glmRetention)
```

```
##
## Call:
## glm(formula = Churn ~ ., family = binomial, data = retention)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9413  -0.7670  -0.3176   0.8287   3.3350
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       2.4719049  1.6185456   1.527 0.126702
## genderMale                       -0.0046624  0.0773545  -0.060 0.951938
## SeniorCitizenYes                  0.1724506  0.0964007   1.789 0.073632 .
## PartnerYes                       -0.1052424  0.0913963  -1.151 0.249529
## DependentsYes                    -0.0063856  0.1079310  -0.059 0.952822
## tenure                           -0.0824664  0.0108992  -7.566 3.84e-14 ***
## MultipleLinesYes                  0.6550271  0.1988972   3.293 0.000990 ***
## InternetServiceFiber optic        2.3899478  0.8948486   2.671 0.007567 **
## OnlineSecurityYes                -0.1198141  0.2011133  -0.596 0.551340
## OnlineBackupYes                   0.1119171  0.1962648   0.570 0.568518
## DeviceProtectionYes               0.1943728  0.1988872   0.977 0.328420
## TechSupportYes                   -0.0312166  0.2028606  -0.154 0.877703
## StreamingTVYes                    0.7301937  0.3652380   1.999 0.045584 *
## StreamingMoviesYes                0.7529349  0.3664171   2.055 0.039893 *
```

```
## ContractOne year                      -0.5604330  0.1327465  -4.222 2.42e-05 ***
## ContractTwo year                      -1.2413921  0.2225390  -5.578 2.43e-08 ***
## PaperlessBillingYes                    0.3101538  0.0905472   3.425 0.000614 ***
## PaymentMethodCredit card (automatic)   0.0879415  0.1332406   0.660 0.509241
## PaymentMethodElectronic check          0.4039426  0.1091998   3.699 0.000216 ***
## PaymentMethodMailed check              0.0522753  0.1432352   0.365 0.715140
## MonthlyCharges                        -0.0642192  0.0356363  -1.802 0.071534 .
## TotalCharges                           0.0005249  0.0001177   4.461 8.16e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5475.1  on 4331  degrees of freedom
## Residual deviance: 4042.1  on 4310  degrees of freedom
##   (3 observations deleted due to missingness)
## AIC: 4086.1
##
## Number of Fisher Scoring iterations: 6
```

(b) There are four payment methods available for customers.

```
levels(retention$PaymentMethod)
```

```
## [1] "Bank transfer (automatic)" "Credit card (automatic)"
## [3] "Electronic check"          "Mailed check"
```

While holding other predictors in the model constant, which payment method category is associated with the largest retention probability? Uncomment your answer below (ctrl-shift-c/cmd-shift-c).

Electronic check

Which payment method category is associated with the smallest retention probability? Uncomment your answer below (ctrl-shift-c/cmd-shift-c).

Bank transfer (automatic)

What is the probability difference comparing the payment method category with largest retention probability to that with the smallest? Uncomment your answer below (ctrl-shift-c/cmd-shift-c).

E. Not enough information for calculating the difference.

(c) Using your fitted model, generate predictions on the test set `retention.test`. What is the test set prediction accuracy (ie the proportion you got right)?
Hint. Use the `predict()` function with argument `type = "response"` to get the predicted probabilities. When the probability is larger than 0.5, our prediction is 1.

```
#  your code here
test.glm = glm(Churn ~ ., family = binomial, data = retention.test)
prediction <- predict(test.glm, newdata = retention.test, type = "response")
prediction <- ifelse(prediction > 0.5,1,0)
accuracy <- mean(prediction == retention.test$Churn)
accuracy
```

```
## [1] 0.732
```

Answer here: Accuracy is 0.732