Homework 9

Maya Mileva

due date: Dec 6th, 2019

I did this homework by myself, with help from the book and the professor.

Exercises

1. The built-in data sets of R include one called "mtcars," which stands for Motor Trend cars. Motor Trend was the name of an automotive magazine and this data set contains information on cars from the 1970s. Use "?mtcars" to display help about the data set. The data set includes a dichotomous variable called vs, which is coded as 0 for an engine with cylinders in a v-shape and 1 for so called "straight" engines. Use logistic regression to predict vs, using two metric variables in the data set, gear (number of forward gears) and hp (horsepower). Interpret the resulting null hypothesis significance tests.

```
## Display help about the data set
# ?mtcars
mycars <- mtcars
dim(mycars)
```

```
## [1] 32 11
```

```
# str(mycars)
```

The data set comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

[, 1] mpg Miles/(US) gallon

[, 2] cyl Number of cylinders

[, 3] disp Displacement (cu.in.)

[, 4] hp Gross horsepower

[, 5] drat Rear axle ratio

[, 6] wt Weight (1000 lbs)

[, 7] qsec 1/4 mile time

[, 8] vs Engine (0 = V-shaped, 1 = straight)

[, 9] am Transmission (0 = automatic, 1 = manual)

[,10] gear Number of forward gears

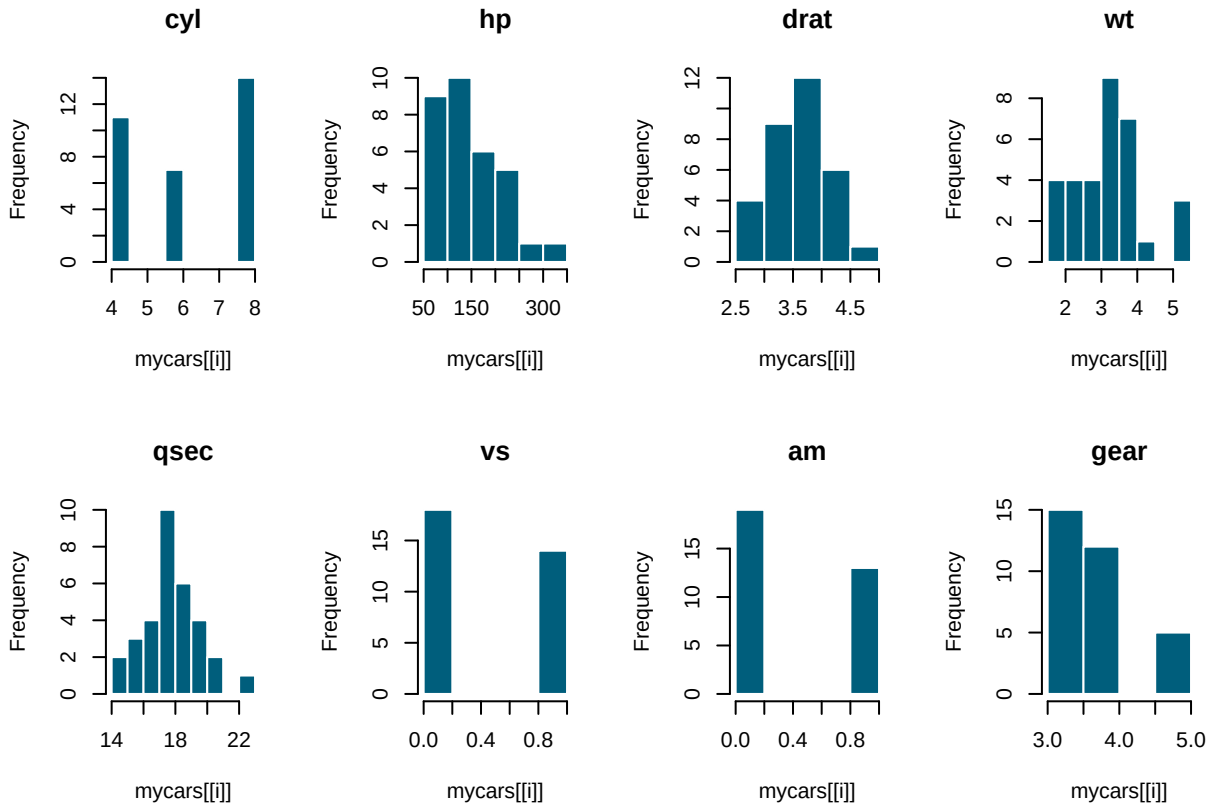
[,11] carb Number of carburetors

```
kable(head(mtcars),align = 'c')
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

The data set includes a dichotomous variable called vs, which is coded as **0** for an engine with cylinders in a v-shape and **1** for so called "straight" engines.

```
par(mfrow=c(2,4))
for (i in c(2,4:10)) hist(mycars[[i]],main=colnames(mycars)[i], col = "#005E7C", border ="white")
```



We have to use logistic regression to predict vs, using two metric variables in the data set, gear (number of forward gears) and hp (horsepower).

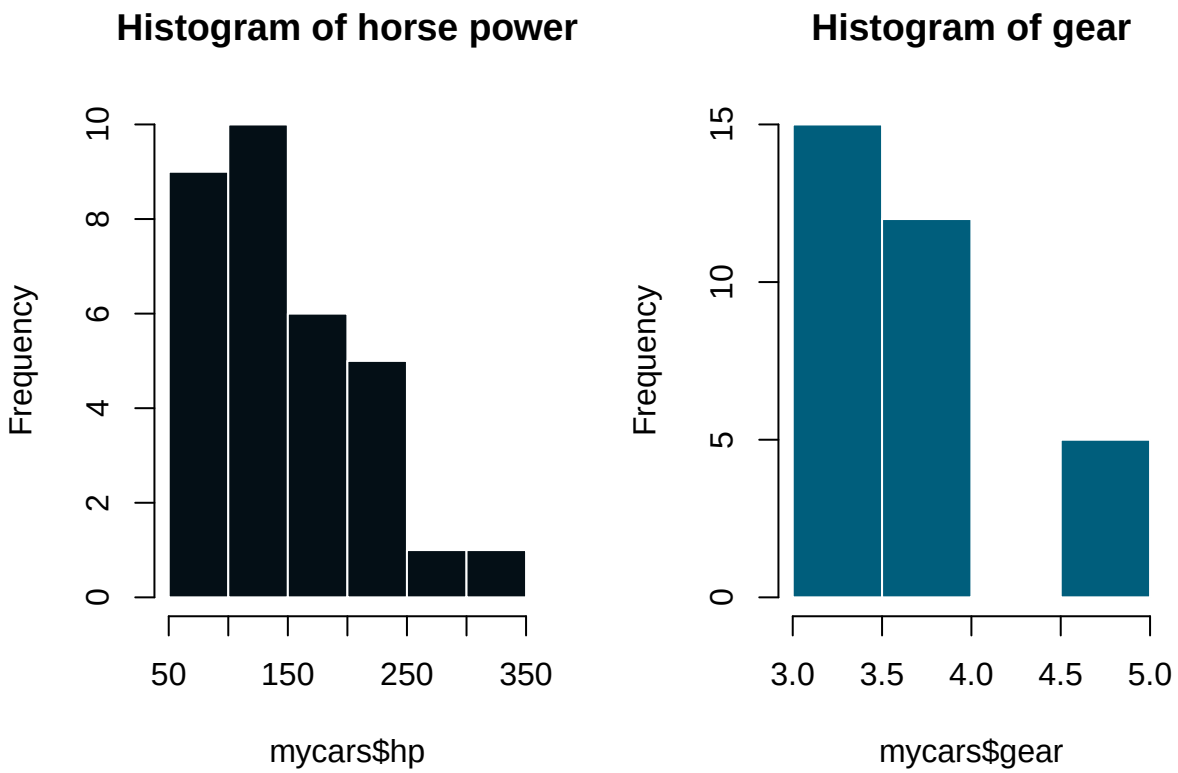
```
round(cor(mycars[,c(2,4:10)]),2)
```

	cyl	hp	drat	wt	qsec	vs	am	gear
cyl	1.00	0.83	-0.70	0.78	-0.59	-0.81	-0.52	-0.49
hp	0.83	1.00	-0.45	0.66	-0.71	-0.72	-0.24	-0.13
drat	-0.70	-0.45	1.00	-0.71	0.09	0.44	0.71	0.70

	cyl	hp	drat	wt	qsec	vs	am	gear
wt	0.78	0.66	-0.71	1.00	-0.17	-0.55	-0.69	-0.58
qsec	-0.59	-0.71	0.09	-0.17	1.00	0.74	-0.23	-0.21
vs	-0.81	-0.72	0.44	-0.55	0.74	1.00	0.17	0.21
am	-0.52	-0.24	0.71	-0.69	-0.23	0.17	1.00	0.79
gear	-0.49	-0.13	0.70	-0.58	-0.21	0.21	0.79	1.00

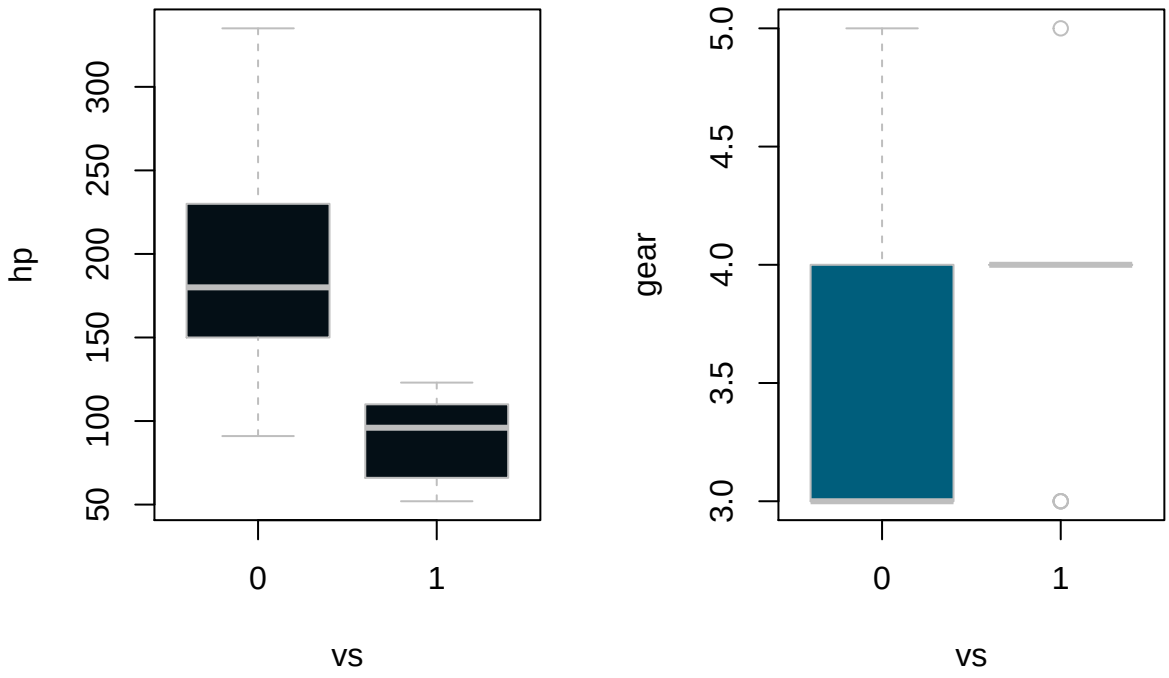
We can see strong negative correlation between "vs" and "hp". Correlation between "gear" and "hp" is .21, which is small and not strong. "hp" and "gear" are not strongly correlated either.

```
par(mfrow = c(1,2))
hist(mycars$hp, main =  "Histogram of horse power",
     col = "#040F16", border = "white")
hist(mycars$gear, main =  "Histogram of gear",
     col = "#005E7C", border = "white")
```



Both predictors are not normally distributed. The range for "hp" is from approximately 50 to 350, and the range for gear varies from 3 to 5.

```
mycars$vs <- as.factor(mycars$vs)
par(mfrow=c(1,2))
boxplot(hp~vs, data = mycars, col = "#040F16", border = "gray")
boxplot(gear~vs, data = mycars, col = "#005E7C", border = "gray")
```



The Distribution of grear has outliers. It looks like mostly only gear 4 has straight engine. There is some overlaping. So the qestion here if "hp" going to be a good predictor.

GLM Output

```
glmOut <- glm(formula = vs ~ hp + gear,
              family = binomial(link="logit"),
              data = mycars)
summary(glmOut)
```

```
##
## Call:
## glm(formula = vs ~ hp + gear, family = binomial(link = "logit"),
##     data = mycars)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.76095  -0.20263  -0.00889   0.38030   1.37305
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 13.43752    7.18161   1.871   0.0613 .
## hp          -0.08005    0.03261  -2.455   0.0141 *
## gear        -0.96825    1.12809  -0.858   0.3907
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.860  on 31  degrees of freedom
## Residual deviance: 16.013  on 29  degrees of freedom
## AIC: 22.013
##
## Number of Fisher Scoring iterations: 7
```

In the equation we can see the "link function" - in this case indicating binomial(). By specifying "binomial()" we invoke the inverse logit as the basis of fitting the X variables("hp" and "gear") to the Y variable ("vs").

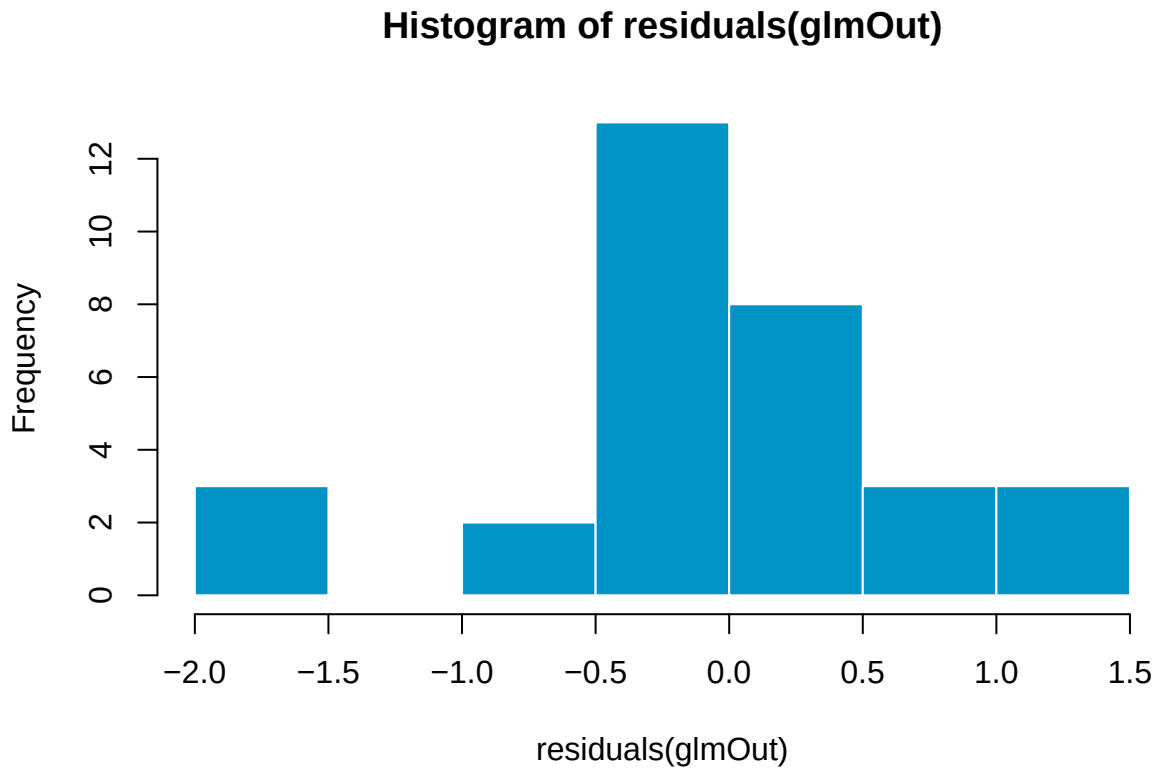
The "Deviance Residuals" show diagnostic information about the distribution of the residuals after the model is fit. The mean of the residuals should be always 0, in our case slightly over 0.

```
mean(residuals(glmOut))
```

```
## [1] 0.0149602
```

The fact that the median residual is slightly negative suggest that the distribution of the residuals is slightly positive skewed.

```
hist(residuals(glmOut), col = "#0094C6", border = "white")
```



These residuals represent error of prediction. If there is residual that is strongly positive or strongly negative, it might suggest problem, such as present of an outlier like in this case.

The output shows that the **intercept** is *not significantly different from 0*. The value of the intercept is not very meaningful to us, but we must keep it in the model to make sure that other coefficient are calibrated correctly.

The coefficient on the **"hp"** predictor is *statistically significant*, based on the Wald's z-test value of -2.46 and the associated p-value. Because p-value (0.0141) <.05 we can reject the null hypothesis that the log-odds of "hp" is 0 in the population. The Wald's z-test is calculated by dividing the coefficient value by the standard error.

The tiny coefficient of **"gear"** is *not significantly different from 0*, based on a Wald's z-test value of -0.86 and associated p-value of 0.3907. Thus we *fail to reject* the null hypothesis that the log-odds of "gear" is equal to 0 in the population.

All these coefficients are log-odds values, we need to convert them to regular odds for easier interpretation.

```
confint(glmOut)
```

	2.5 %	97.5 %
(Intercept)	2.2528323	32.536254
hp	-0.1683153	-0.033567
gear	-3.7010673	1.056678

```
exp(cbind(OR = coef(glmOut), confint(glmOut)))
```

	OR	2.5 %	97.5 %
(Intercept)	6.852403e+05	9.5146461	1.349944e+14
hp	9.230734e-01	0.8450873	9.669901e-01
gear	3.797461e-01	0.0246972	2.876797e+00

```
# round(9.669901e-01, digits = 2) #0.97 - for hp 0.85 to 0.97
# round(2.876797e+00, digits = 2) #2.88 - for gear 0.02 to 2.88 # straddle with 1
```

We usually ignore odds ratios for intercept terms. The odds of having a straight, rather than v-shaped engine, increase by 0.92 (rounding up 9.230734e-01) for every unit increase in gross horsepower. In this case the odds are almost 1:1 (the odds of "1" do not change at all in response of changing horsepower). The 95% confidence interval for "hp" ranged from 0.84:1 up to 0.97:1 (really close to 1, but doesn't straddle), expressed in plain odds; if the study was repeated 100 times, 95% of similarly constructed intervals would contain the true population value.

The confidence interval for "gear" stradles with 1:1, confirming the nonsignificant results for that coefficient. The definition of CI is that if you constructed a very large number of similar experiments based on new samples, 95% of the CI you would calculate would contain the population value.

We have to take a look at the deviance too.

Null deviance: 43.860 on 31 degrees of freedom

Residual deviance: 16.013 on 29 degrees of freedom

AIC: 22.013

Number of Fisher Scoring iterations: 7

The last line shows how many iterations of the model fitting it took to get the final model. AIC is a measure of stress in the model. The "Null Deviance" shows amount of error in the model, if we pretend there is no connection between X variables and Y variable. It shows what would happen if the predictors had no predictive value. The null model shows 31 degrees of freedom for calculating the proportion of straight and v-shaped engine. The null model in some ways represents the null hypothesis. The next line shows how much error is reduced by introducing the X variables. We lose 2 degrees by introducing 2 variables. By introducing 2 predictors we reduced error from 43.860 to 16.013 (which cost 3 degree of freedom).

The difference between the null model and the residual model is distributed as chi-square and can be used as an omnibus test.

```
# Compare null model to two predictor model
anova(glmOut, test = "Chisq")
```

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL	NA	NA	31	43.86011	NA
hp	1	27.0225283	30	16.83758	0.0000002
gear	1	0.8244408	29	16.01314	0.3638850

The first chi-square test shows a difference of 27.0225 on the one degree of freedom for model with just "hp" as predictor which is statistically significant. The second chi-square test of 0.8244 had "hp" and "gear" as predictors and is not statistically significant. Adding the second predictor didn't segnificantly reduced the residual deviance. These results make sense in the light of the significance test on the coefficients and confirms the utility of a model that contains only "hp".

```
table(round(predict(glmOut, type= "response")), mycars$vs)
```

/	0	1
0	15	3
1	3	11

The off diagonal items, 3 and 3 are all the errorneous predictions.

```
## Overall accuracy
(15+11)/32
```

```
## [1] 0.8125
```

```
(3+3)/32 # error rate
```

```
## [1] 0.1875
```

The overall accuracy was 81%.

Conclusion

We tested a measures of horse power ("hp", ranged from 30 to 350) and gear("gear", range from 3 to 5) to see if they could predict the shape of the can engine("vs"). A chi-square omnibus test on the result of logistic

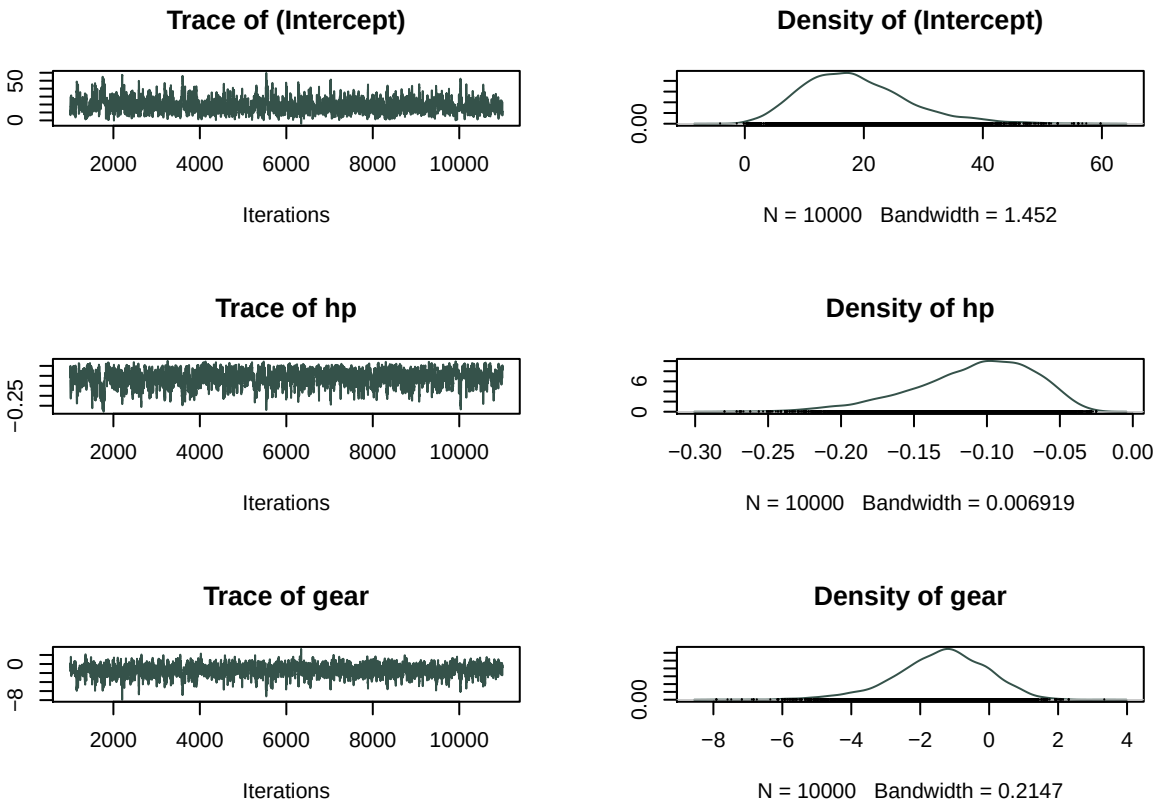
regression was significant for model with the one predictor, chisq(1) = 27.0225, p<.0001. Only the Wald's z-test on the "hp" coefficient was significant, z= -2.46, p<.05. When converted to odds, the coefficient was 0.92 suggesting that for each unit increase in horsepower, the odds of the engine being straight increased by .92:1. I cannot say that this is a strong evidence that horsepower could serve as useful "vs" predictor. The HDI for "hp" in plain odds is 0.85 to 0.97.

Bayesian Analysis

```
mycars$vs <- as.numeric(mycars$vs)- 1
bayesLogitOut <- MCMClogit(formula = vs ~ hp+gear, data = mycars)
summary(bayesLogitOut)
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                Mean      SD  Naive SE Time-series SE
## (Intercept) 18.5047 8.82570 0.0882570       0.339443
## hp          -0.1087 0.04119 0.0004119       0.001671
## gear        -1.3983 1.34302 0.0134302       0.047170
##
## 2. Quantiles for each variable:
##
##                2.5%     25%    50%      75%    97.5%
## (Intercept)  4.4716 12.0818 17.500 23.66161 39.08301
## hp          -0.2042 -0.1337 -0.103 -0.07784 -0.04555
## gear        -4.4295 -2.1901 -1.306 -0.47733  0.93486
```

```
plot(bayesLogitOut, col = "#35524A", border = "white")
```



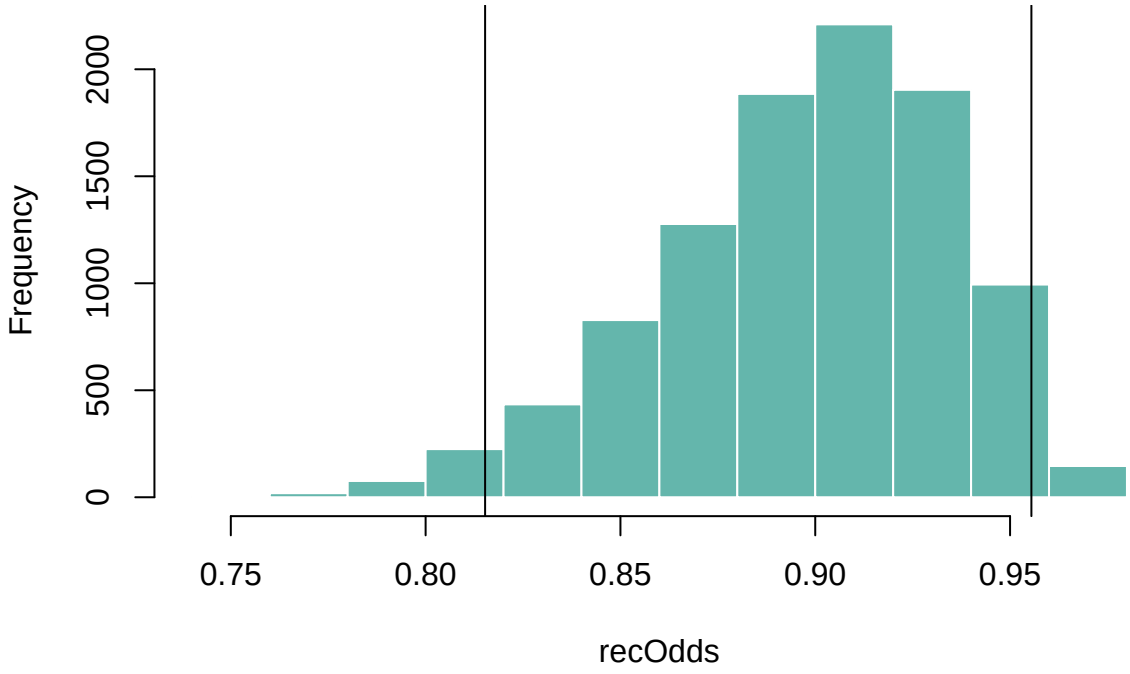
Trace plots show the progress of the MCMC estimation process. Density plots show the posterior distribution of each coefficient. "gear" is centered near 0 which confirm that there isn't much going on with that variable and might not be a good predictor. They are all quite normally shaped and the central region of 95% under the curve is where in all likelihood the parameter of interest lies.

The output of MCMC focuses on describing the distribution of the parameters representing both the intercept and the coefficients of "hp" and "gear", calibrated in log-odds.

The point estimates in the current output are listed under the "Mean" column and are similar to the output from the traditional logistic regression. The next column "SD" corresponds to the standard error in the output. The most common points of interest will be the log-odd coefficients of the two predictors. One of them is for "hp" -0.1087 and the other is "gear" -1.39. In the second output we can clearly see that the HDI for "gear" overlap with 0, so the population parameter for "gear" lies somewhere near 0. We need to convert "hp" to plain odds in order to interpret it, because the iinterval does not overlap with 0 and we can use that predictor.

We can improve our view of the parameter estimates of the coefficient by converting the distribution from log odds to plain odds.

```
recLogOdds <- as.matrix(bayesLogitOut[,"hp"])
recOdds <- apply(recLogOdds,1,exp)
hist(recOdds, main=NULL, col = "#64B6AC", border = "white")
abline(v=quantile(recOdds,probs=c(0.025,0.975)))
```




```
# mean(recOdds) # 0.8977491
```

The histogram shows a distribution centered around .88 consistent with the results we obtained from glm(). The HDI bounds here are similar to but not identical to the confidence interval from glm() too. Odds of 0.89:1 for the coefficient on the "hp", don't make any big changes.

Conclusion

We examined the data from the 1974 Motor Trend US magazine, which comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles, and tried to see if the horsepower and gear of a car could predict wheatear the car would have straight of v-shaped engine. We conducted Bayesian logistic analysis, using horsepower and gear to predict engine. The posterior distribution of the coefficient for gear (calibrated as log odds) overlapped squarely with 0, suggesting that gear was not meaningful predictor of engine. In contrast the HDI for horsepower did not overlap with zero. When converted to regular odd, the mean value of the posterior distribution for hp was .89:1 suggesting that for every additional horsepower, car is about 1% more likely to have v=shaped engine. The confusion matrix showed overall error rate of 18% indicating that the logistic model was somehow good but the power of "hp" was a predictor was not that strong.

5. As noted in the chapter, the BaylorEdPsych add-in package contains a procedure for generating pseudo-R-squared values from the output of the glm() procedure. Use the results of Exercise 1 to generate, report, and interpret a Nagelkerke pseudo-R-squared value.

```
PseudoR2(glmOut)
```

```
##         McFadden     Adj.McFadden        Cox.Snell       Nagelkerke
```

```
##        0.6349042        0.4525061        0.5811397        0.7789526
## McKelvey.Zavoina           Effron            Count        Adj.Count
##        0.8972195        0.6445327        0.8125000        0.5714286
##              AIC    Corrected.AIC
##       22.0131402       22.8702830
```

R- square summarizes the overall goodness of the model.

A pseudo R-squared only has meaning when compared to another pseudo R-squared of the same type, on the same data, predicting the same outcome. The interpretation is: "the proportion of the total variability of the outcome that is accounted for by the model".In this situation, the higher pseudo R-squared indicates which model better predicts the outcome.

Cox and Snell's(.58) is based on the log likelihood for the model compared to the log likelihood for a baseline model. However, with categorical outcomes, it has a theoretical maximum value of less than 1, even for a "perfect" model. Nagelkerke is an adjusted version of the Cox & Snell R-square that adjusts the scale of the statistic to cover the full range from 0 to 1. The model with the largest R square statistic is "best" according to this measurelike we said earlier.

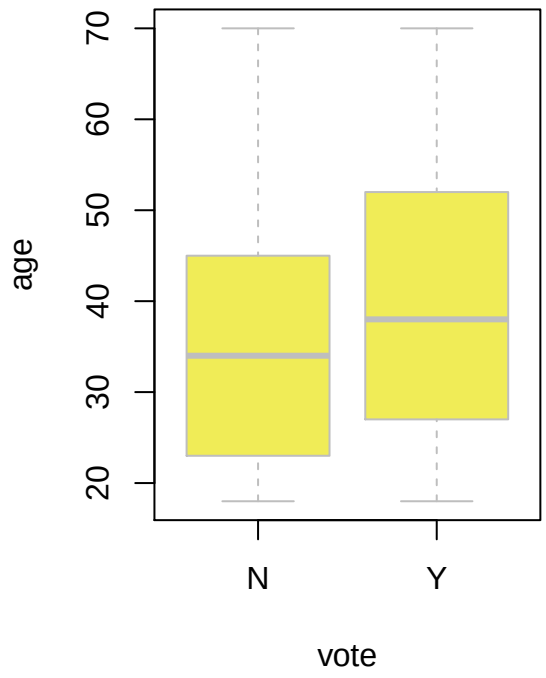
If we examine Nagelkerke(0.7789526) we can interpret it as the proportion of variance in the outcome variable("vs") accounted by the predictor variables ("hp" and "gear"). The value is good for R-square (really close to 1). Given out that only "hp" as significant, these results suggest that "hp" has moderate role for accounting for the "vs" variable. This returns us to the idea of the statistical power.

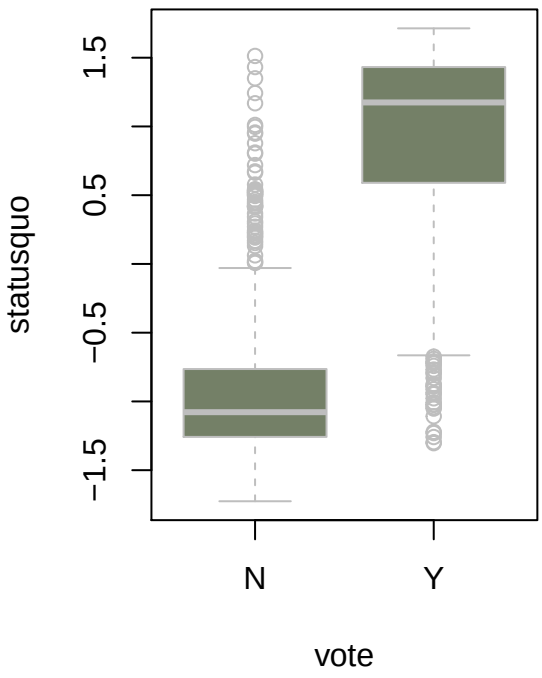
6. Continue the analysis of the Chile data set described in this chapter. The data set is in the "car" package, so you will have to install.packages() and library() that package first, and then use the data(Chile) command to get access to the data set. Pay close attention to the transformations needed to isolate cases with the Yes and No votes as shown in this chapter. Add a new predictor, statusquo, into the model and remove the income variable. Your new model specification should be vote ~ age + statusquo. The statusquo variable is a rating that each respondent gave indicating whether they preferred change or maintaining the status quo. Conduct general linear model and Bayesian analysis on this model and report and interpret all relevant results. Compare the AIC from this model to the AIC from the model that was developed in the chapter (using income and age as predictors).

```
data(Chile)

## Grab Yes votes
ChileY <- Chile[Chile$vote == "Y",]
## Grab No votes
ChileN <- Chile[Chile$vote == "N",]

## Create new data set
ChileYN <- rbind(ChileY, ChileN)
ChileYN <- ChileYN[complete.cases(ChileYN),]

## Simplify the factor
ChileYN$vote <- factor(ChileYN$vote, levels = c("N","Y"))
```

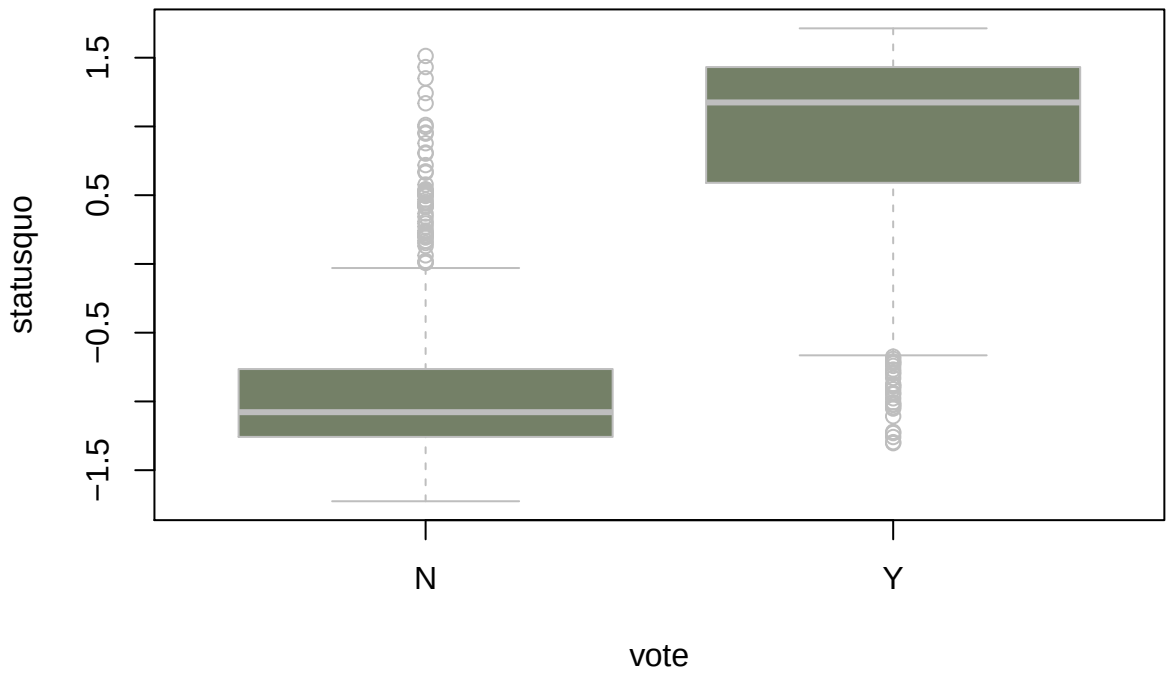
We can create boxplots to represent ranges for each of the predictors, divided by Yes and No votes.

```
par(mfrow = c(1,2))
boxplot(age~vote, data = ChileYN, col = "#F0EC57", border = "gray")
boxplot(statusquo~vote, data = ChileYN, col = "#748067", border = "gray")
```






```
## Displaing all the outliers
# boxplot(statusquo~vote, data = ChileYN,
        # col = "#748067", border = "gray")$out

## Assign the outlier values into a vector
outliers <- boxplot(statusquo~vote, data = ChileYN,
        col = "#748067", border = "gray")$out
```




```
print(outliers)
```

```
##  [1]  1.16851  0.52845  0.71898  0.15214  0.41711  0.02178  0.21063  0.53257
##  [9]  0.41980  0.19175  0.01279  1.01110  0.06239  1.51284  0.43869  1.43203
## [17]  0.95611  0.00720  0.36676  0.24220  0.50186  0.22900  0.57763  0.67327
## [25]  0.51637  0.41409  1.35000  0.99704  0.22416  0.46682  0.34221  0.42371
## [33]  0.27038  0.66365  0.87797  0.80245  0.50129  0.19955  0.13025  0.45050
## [41]  0.81038  0.53171  0.49426  0.51557  0.54349  0.51627  0.94765  0.29437
## [49]  0.30916  1.24288  0.16797 -1.10784 -0.97742 -0.91124 -0.82983 -0.87318
## [57] -1.23032 -0.93686 -1.00723 -1.30351 -0.76832 -0.89149 -0.69048 -0.88666
## [65] -1.05197 -0.72379 -1.29617 -1.25899 -1.02236 -0.71844 -0.67418 -0.79979
## [73] -1.04094 -0.78735 -0.70916 -0.94732 -0.74140 -1.22114 -1.02512
```

```
# ChileYN <- ChileYN[-which(ChileYN$statusquo %in% outliers),]
# boxplot(ChileYN$statusquo)
```

The plot on the right hand suggest that the wealthier voters might be more likely to vote No. For both predictors, there is overlap in the distributions of the predictors for Yes and No votes so it is hard to say wheatear or not this differences are simply due to a sampling error. There are outliers presenting in the boxplot on the right.

Lets use logistic regression to see if we can scientifically predict a Yes or No vote based on the age and statusquo level of a person who responded to the poll:

GLM Output

```
chOut <- glm(formula = vote ~ statusquo + age, family = binomial(),
             data = ChileYN)
summary(chOut)
```

```
##
## Call:
## glm(formula = vote ~ statusquo + age, family = binomial(), data = ChileYN)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2095  -0.2830  -0.1840   0.1889   2.8789
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.193759   0.270708  -0.716   0.4741
## statusquo    3.174487   0.143921  22.057   <2e-16 ***
## age          0.011322   0.006826   1.659   0.0972 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2360.29  on 1702  degrees of freedom
## Residual deviance:  734.52  on 1700  degrees of freedom
## AIC: 740.52
##
## Number of Fisher Scoring iterations: 6
```

In the equation we can see the "link function" - in this case indicating binomial(). By specifying "binomial()" we invoke the inverse logit as the basis of fitting the X variables("age" and "statusquo") to the Y variable ("vote").

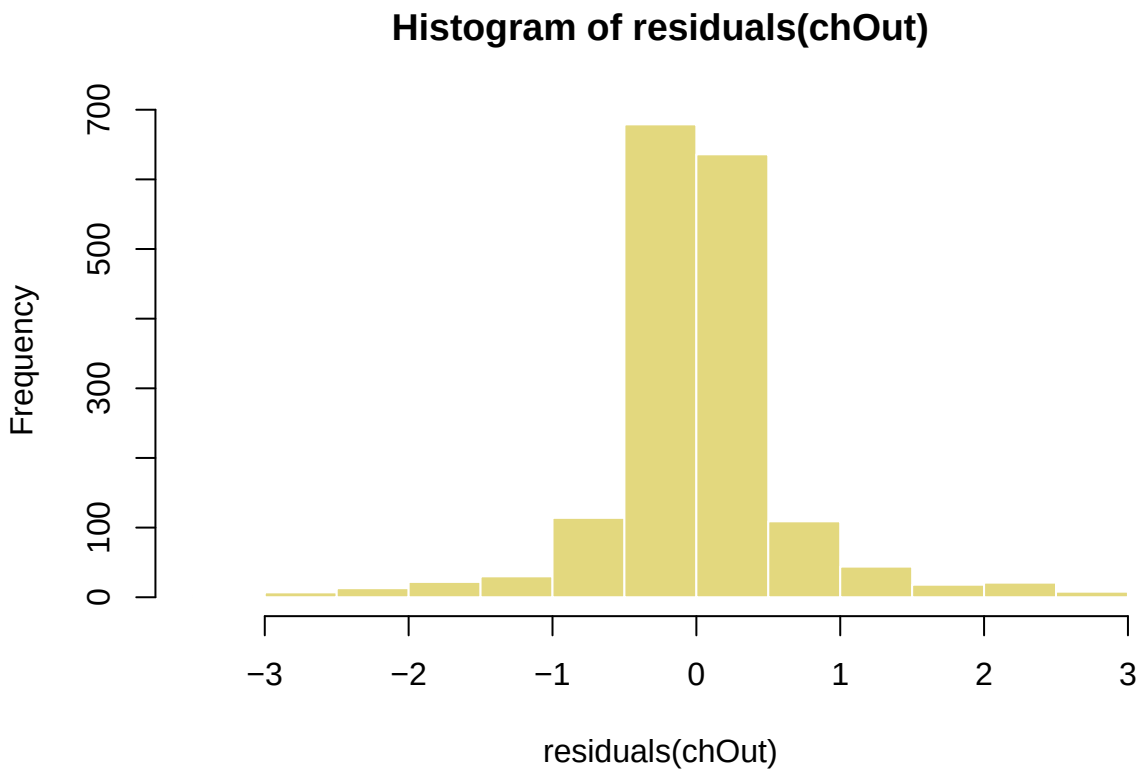
The "Deviance Residuals" show diagnostic information about the distribution of the residuals after the model is fit. The mean of the residuals is always 0 in our case slightly under 0.

```
mean(residuals(chOut))
```

```
## [1] -0.02173092
```

The fact that the median residual is slightly negative suggest that the distribution of the residuals is slightly positive skewed.

```
hist(residuals(chOut), col = "#E3D87E", border = "white")
```



These residuals represent error of prediction. If there is residual that is strongly positive or strongly negative, it might suggest problem, such as present of an outlier.

The output shows that the **intercept** is *not significantly different from 0*. The value of the intercept is not very meaningful to us, but we must keep it in the model to make sure that other coefficient are calibrated correctly.

The coefficient on the **"statusquo"** predictor is *statistically significant*, based on the Wald's z-test value of 22.057 and the associated p-value. Because p-value (2e-16 ***) <.001 we can reject the null hypothesis that the log-odds of "statusquo" is 0 in the population. The Wald's z-test is calculated by dividing the coefficient value by the standard error.

The tiny coefficient of **"age"** is *not significantly different from 0*, based on a Wald's z-test value of 1.659 and associated p-value of 0.0972. Thus we *fail to reject* the null hypothesis that the log-odds of "age" is equal to 0 in the population.

All these coefficients are log-odds values, we need to convert them to regular odds for easier interpretation. In this case we only significant is "statusquo".

```
confint(chOut)
```

	2.5 %	97.5 %
(Intercept)	-0.7242110	0.3385679
statusquo	2.9040747	3.4690947
age	-0.0020686	0.0247244

```
exp(cbind(OR = coef(chOut), confint(chOut)))
```

	OR	2.5 %	97.5 %
(Intercept)	0.8238564	0.4847068	1.402937
statusquo	23.9145451	18.2483505	32.107663
age	1.0113863	0.9979335	1.025033

The intercept represents odds 0f 0.82:1 for Yes vote by somebody without income that prefer to maintain the status quo. The odds of 1.011:1 for age show that every additional year of age, a person is about 1.1% more likely to vote Yes. In the case of statisquo, the odds are 23.91:1, which going to make big difference. These results agree with the hypothesis tests: the confidence interval for age straddles 1:1, confirming non-significant result for that coefficient. The 95% CI for the intercept ranges from 0.48:1 up to 1.40:1 but we don't need to interpret it like we established earlier. The CI of statusquo runs from low 18.24:1 up to 32.10. We have to mation the outliers presenting in the statusquo variable when we are talking about significance and prediction power.

We should calculate and report the results from the chi-square test.

```
anova(chOut, test = "Chisq")
```

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL	NA	NA	1702	2360.2950	NA
statusquo	1	1623.026234	1701	737.2687	0.0000000
age	1	2.748047	1700	734.5207	0.0973733

We have separate tests that compare three "nested" models. The first chi-square test compared the null model to a model that just includes the statusquo predictor. The second chi-square compares the model with just age to a model that has both age and income as predictors. Only the first chi-square is statistically significant (because p = 2e-16 *** is below the threshold of p<.001). These results make sense in the light of the significance test on the coefficients and confirms the utility of a model that contains only statusquo.

Each successive line of the output, we lose a degree of freedom each time we enter a new predictor. The column "Deviance Resid" is the chi-square value for the effect of the predictor, while "Dev" is the chi-square that represent what is unaccounted for in the dependent variable after the entry of each predictor in the model.

To close our consideration of the output of glm(), we will reproduced few lines from the output earlier.

Null deviance: 2360.29 on 1702 degrees of freedom

Residual deviance: 734.52 on 1700 degrees of freedom

AIC: 740.52

Number of Fisher Scoring iterations: 6

The model took 6 iteration in order to produce the final model. The "Null Deviance" shows amount of error in the model, if we pretend there is no connection between X variables and Y variable. It shows what would happen if the predictors had no predictive value. The null model shows 17002 degrees of freedom for calculating the proportion of Yes and No votes. The null model in some ways represents the null hypothesis. The next line shows how much error is reduced by introducing the X variables. We lose 2 degrees by introducing 2 variables. By introducing 2 predictors we reduced error from 2360.29 to 734.53 (which cost 3 degree of freedom) but is really great reduction. The difference between the null model and the residual model

is distributed as chi-square and can be used as an omnibus test. Another note is about the overall result is about AIC. AIC stands for Akaike information criterion and examines the error reduction accomplished by a model considering the number of parameters. If we want to compare the results from our model (AIC = 740.52), and model that predict vote from income and age AIC (2332), we will choose the model with the lowest AIC. It is taking into account the number of predictors, but in our case, they are two for both models.

```
table(round(predict(chOut, type= "response")), ChileYN$vote)
```

/	N	Y
0	810	74
1	57	762

The off diagonal items, 54 and 74 are all the errorneous predictions.

```
## Overall accuracy
(810+762)/(810+74+57+762) #92%
```

```
## [1] 0.9230769
```

```
(74+57)/(810+74+57+762) # error rate 8%
```

```
## [1] 0.07692308
```

We can say that is a really good model.

We tested a measures of age and statusquo to see if they could predict the vote of people in Chile. A chi-square omnibus test on the result of logistic regression was significant for model with the one predictor, chisq (1) = 1623.03, p<.0001. Only the Wald's z-test on the statusquo coefficient was significant, z= 22.057, p <.05. When converted to odds, the coefficient was 23.91 suggesting that for each individual that who wants to maintain statusquo, the odds of that person voting Yes is 23.91:1. This is a strong evidence suggesting that statusquo could serve az a useful vote predictor. The 95% CI for statusquo range from low 18.24:1 up to 32.10, expressed in plain odd. If the study was represented 100 times, 95% of the similarly constructed intervals would contain the population value.

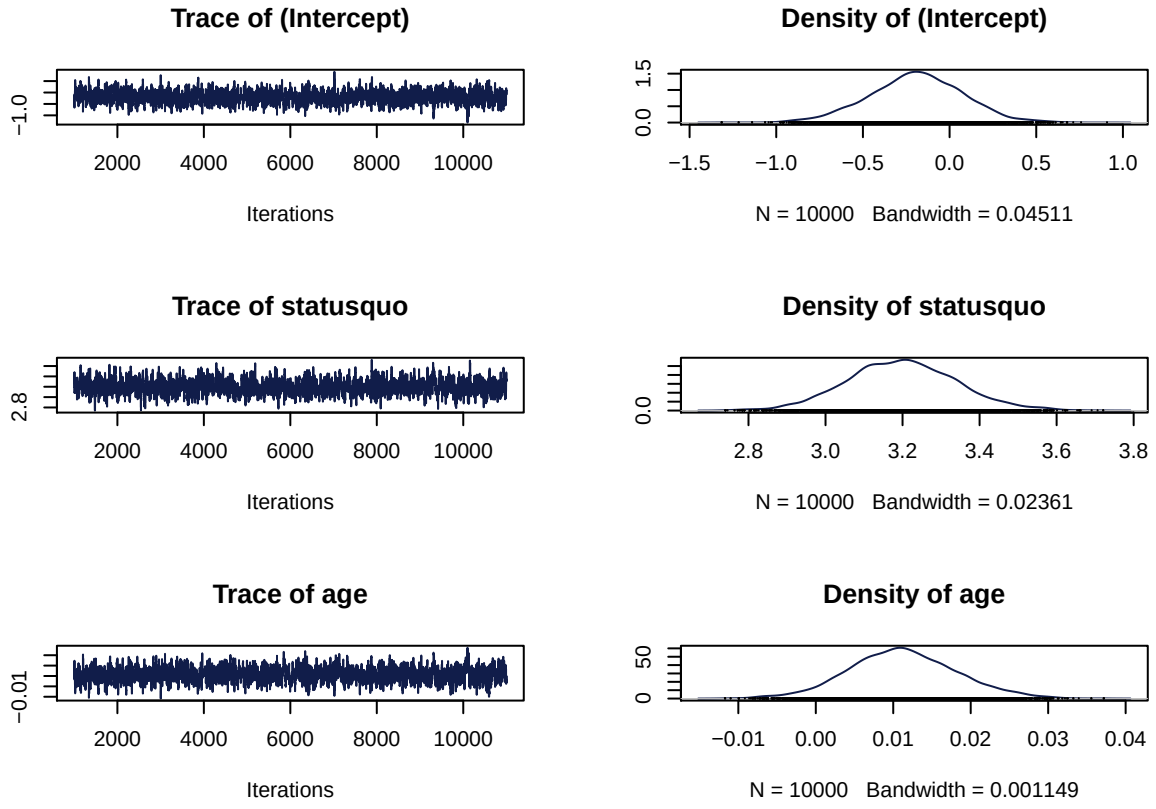
Bayesian Analysis

```
## Adjust the outcome variable
ChileYN$vote <- as.numeric(ChileYN$vote)- 1

bayesLogitOut1 <- MCMClogit(formula = vote ~ statusquo + age, data = ChileYN)
summary(bayesLogitOut1)
```

```
##
## Iterations = 1001:11000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
```

```
##    plus standard error of the mean:
##
##                  Mean       SD  Naive SE Time-series SE
## (Intercept) -0.19230 0.270280 2.703e-03      0.0088240
## statusquo    3.19813 0.141422 1.414e-03      0.0046907
## age          0.01123 0.006917 6.917e-05      0.0002295
##
## 2. Quantiles for each variable:
##
##                 2.5%       25%      50%       75%   97.5%
## (Intercept) -0.73688 -0.367807 -0.18743 -0.007969 0.32179
## statusquo    2.91721  3.104102  3.19802  3.292425 3.48629
## age         -0.00235  0.006636  0.01107  0.015803 0.02556
```

```
plot(bayesLogitOut1, col = "#111D4A")
```



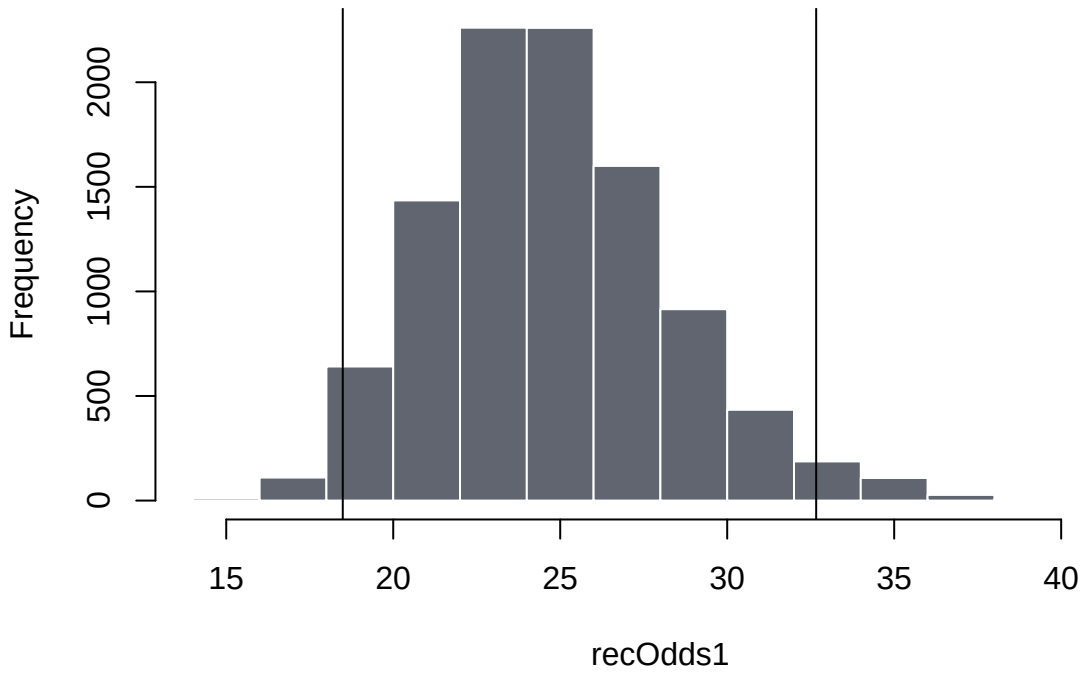
Trace plots show the progress of the MCMC estimation process. Density plots show the posterior distribution of each coefficient. "age" is centered near 0 which confirm that there isn't much going on with that variable and might not be a good predictor. Intercept also centers around 0. They are all quite normally shaped and the central region of 95% under the curve is where in all likelihood the parameter of interest lies.

The output of MCMC focuses on describing the distribution of the parameters representing both the intercept and the coefficients of age and statusquo, calibrated in log-odds.

The mean value of each coefficient is the "point estimate" at the center of the density distribution. These are fairly close to the output of glm() - 3.19 for statusquo and 0.001 for age. The next column "SD" corresponds to the standard error in the output. In the second output we can clearly see that the HDI for age overlap

with 0, so the population parameter for "age" lies somewhere near 0. We need to convert statusquo to plain odds in order to interpret it, because the interval does not overlap with 0 and we can use that predictor.

```
recLogOdds1 <- as.matrix(bayesLogitOut1[,"statusquo"])
recOdds1 <- apply(recLogOdds1,1,exp)
hist(recOdds1, main=NULL, col = "#60656F", border = "white")
abline(v=quantile(recOdds1,probs=c(0.025,0.975)))
```




```
mean(recOdds1) # 24.73367
```

```
## [1] 24.73367
```

The histogram shows almost symmetric distribution centered about 24.73, consistent with the results that we obtained from glm() and suggesting an increase of about 24.73:1 in likelihood of Yes vote. The HDI spans a region starting at as low as 18.49 and ranging as high as 32.66. These boundaries are really close to those of the CI we obtained from glm(). HDI gives us a direct view of the most likely range of coefficient values in the population.

We examined the Chile vote data and the research question was whether age and statusquo predicted Chileans' votes on a plebiscite in 1988. Voting "Yes" was a vote to keep then-president Augusto Pinochet in office. We conducted a Bayesian logistic analysis, using age and statusquo to predict votes. The posterior distribution of the coefficient for age (calibrated as log odds) overlapped squarely with zero, suggesting that age was not a meaningful predictor of votes. In contrast, the Highest Density Interval of statusquo did not overlap with zero. When converted to regular odds, the mean value of the posterior distribution for statusquo was 23.91 to 1, suggesting that for every additional status meantain, an individual was about 24%

more likely to vote to keep Pinochet. However, a confusion matrix showed that the overall error rate was 8% indicating that the logistic model was particularly good at predicting votes.

7. Bonus R code question: Develop your own custom function that will take the posterior distribution of a coefficient from the output object from an MCMClogit() analysis and automatically create a histogram of the posterior distributions of the coefficient in terms of regular odds (instead of log-odds). Make sure to mark vertical lines on the histogram indicating the boundaries of the 95% HDI.

```
logFuncHist <- function(logO){
 statusQuoLogOdds <- as.matrix(logO[,"statusquo"])
 statusOdds <- apply(statusQuoLogOdds,1,exp)
 hist(statusOdds, col = "#f7c297", border = "white")
 abline(v=quantile(statusOdds,c(.025)),col='black')
 abline(v=quantile(statusOdds,c(.975)),col='black')
}
logFuncHist(bayesLogitOut1)
```