Exercises from: Modeling correlated, clustered responses

Chapter 9 Agresti (2007)

Saúl Sotomayor Leytón

May 2012

Setup

```
> library(vcd)
> library(survival)
> library(exact2x2)
> library(BradleyTerry2)
> library(gee)
> library(geepack)
> library(MASS)
> options(width=70)
```

Problem 1

a) To calculate the proportions of those who consume alcohol (A), cigarette (C) and marijuana (M) we use the code from exercise 8.12. Note however that in the original code was a mistake in calculating the proportion of those who consume marijuana.

```
> tb7.3 <- data.frame(expand.grid(cigarrette=c('yes', 'no'),
+                                 alcohol=c('yes', 'no'),  marijuana=c('yes', 'no')),
+                       count=c(911, 44, 3, 2, 538, 456, 43, 279))
> tb.ex1A <- xtabs(count ~ alcohol + cigarrette, data = tb7.3)
> tb.ex1C <- xtabs(count ~ cigarrette + alcohol, data = tb7.3)
> tb.ex1M <- xtabs(count ~ marijuana + alcohol, data = tb7.3)
```

Once the data is constructed we calculate the proportions with the functions, `margin.table` and `prop.table`, the latter transforms the counts into proportions and the former calculates the margins of the table.

```
> round(margin.table(prop.table(tb.ex1A), 1)[2], 3)
```

```
   no
0.144
```

```
> round(margin.table(prop.table(tb.ex1C), 1)[2], 3)
```

```
   no
0.343
```

```
> round(margin.table(prop.table(tb.ex1M), 1)[2], 3)
```

```
   no
0.578
```

In the `margin.table` function the number `1` indicates that we want the row margins, while the index, `[2]`, indicates that we want only the second row, that corresponding to the yes proportion.

b) A marginal model to compare the margins for the different drugs could be one that uses a dummy variable for each one.

$$logit[P(Y_t = 1)] = \alpha + \beta_1 A + \beta_2 C + \beta_3 M$$

Note that $P(Y_t = 1)$ is the probability of consuming substance t, where A=1 when t=1, C=1 when t=2, and M=1 when t=3. In this model, $exp(\beta_1)$ would be the odds of using alcohol, and a marginal homogeneity hypothesis would be, $H_0 : \beta_1 = \beta_2 = \beta_3$. Remember from Agresti (2007, p. 242) that marginal homogeneity means that the proportions for a particular outcome are equal for every classification (*e.g.* substance's use or question type).

Problem 2

In order to fit a GEE model for the data in table 7.13 we need to transform it. First we need to collapse the columns for each drug into a single one; this is done with the `reshape` function.

```
> tb7.13 <- data.frame(expand.grid(C=c(1, 0), A=c(1, 0), R=c('W', 'O'),
+                                  G=c('F', 'M'), M=c(1, 0)),
+                      Count=c(405, 13, 1, 1, 23, 2, 0, 0, 453, 28, 1, 1, 30, 1, 1, 0,
+                              268, 218, 17, 117, 23, 19, 1, 12, 228, 201, 17, 133, 19,
+                              18, 8, 17))
> tb7.13b <- reshape(tb7.13, varying = c("A", "C", "M"), direction = "long",
+     v.names = "outcome", timevar = "drug")
> tb7.13b$drug <- ifelse(tb7.13b$drug == 1, "alcohol", ifelse(tb7.13b$drug ==
+     2, "cigarrette", "marijuana"))
> tb7.13b$drug <- factor(tb7.13b$drug, levels = c("marijuana",
+     "cigarrette", "alcohol"))
```

Note that the variables in the `varying` argument will be coded in the order they are entered, in this case the `alcohol` variable will become 1, while the `cigarrette` variable will become 2. To avoid confusion, these are renamed in the last command.
The next step is to expand the table into one where each row represents one subject and create a vector (`case`) that will represent the grouping

```
> tb7.13long <- tb7.13b[rep(1:nrow(tb7.13b), tb7.13b$Count), ]
> tb7.13long$case <- rep(1:2276, 3)
> tb7.13long <- tb7.13long[order(tb7.13long$case), ]
```

The number `2276` is the sum of all the counts in the original table, *i.e.* `tb7.13`. Also note that the `case` vector **must** be ordered.
Now we can fit the model, but before that we need to verify that the coding matches that used in Agresti (2007) and also we can delete the, now, unnecessary variables. For the first case, remember that `R` sets to zero the first variable, thus those variables that appear first in the output of the `summary` function are those that will be set to zero. Latter we fit the model with the `gee` function from the library with the same name

```
> tb7.13long$G <- factor(tb7.13long$G, levels = c("M", "F"))
> tb7.13long <- within(tb7.13long, {
```

```
+     rm(Count, id)
+ })
> fit.gee2 <- gee(outcome ~ R + G + drug, id = case, family = binomial,
+     corstr = "exchangeable", data = tb7.13long)
```

```
   (Intercept)             RO             GF drugcigarrette
   -0.26419702    -0.40719846    -0.04333689     0.96759336
   drugalcohol
    2.10622159
```

The coefficients and its standard errors are extracted with the `summary` function but are better expressed in a LaTeX 2_ε table. These values can be interpreted as follows. The odds for a white person to

Table 1: Estimates and its standard errors for the main effects model for table 7.13

	Estimate	Naive S.E.	Naive z	Robust S.E.	Robust z
(Intercept)	-0.67	0.11	-6.37	0.11	-6.32
RW	0.41	0.10	4.07	0.10	4.04
GF	-0.04	0.05	-0.79	0.05	-0.79
drugcigarrette	0.97	0.06	15.77	0.06	15.77
drugalcohol	2.11	0.07	28.68	0.07	28.71

consume at least one of the drugs is $exp(0.41) = 1.51$ times the odds for a non-white person. Similarly the odds for a woman to consume at least one of the drugs is $exp(-0.04) = 0.96$ times the odds of a man to consume the same drugs. Finally the odds of consuming cigarette is $exp(0.97) = 2.64$ times the odds of consuming marijuana and the odds of consuming alcohol are $exp(2.11) = 8.25$ times the odds of consuming marijuana!.

Note on model fitting As it is noted by Agresti a drawback of GEEs is that one can not test model fitting with LR statistics, in the case of `R` with the `anova` function. In this way I don't understand how one can say that an interaction exists between gender and drug type, when, according to the z-value it seems that gender is non-significative.

Problem 3

To fit the interaction model we just change the formula.

```
> fit.gee3 <- gee(outcome ~ R + G * drug, id = case, family = binomial,
+     corstr = "exchangeable", data = tb7.13long)
```

```
     (Intercept)               RO               GF
      -0.1867646       -0.4070079       -0.2021861
   drugcigarrette      drugalcohol GF:drugcigarrette
       0.8618998        1.9279939        0.2161426
   GF:drugalcohol
       0.3686166
```

However it's possible to solve the problem with the equation provided in Agresti,

$$logit(\hat{\pi}) = -0.57 + 1.93A + 0.86C + 0.38W - 0.20F + 0.37F \times A + 0.22F \times C$$

a) For alcohol the greatest value results for the combination, white female: $logit(\hat{\pi}) = -0.57 + 1.93 + 0.38 - 0.20 + 0.37 = 1.91$

b) The odds of a white subject to use a given substance is $exp(0.38) = 1.46$

c) The odds for a female to use alcohol is $exp(-0.20 + 0.37) = exp(0.17) = 1.19$ times that for males. Note that, because we are comparing genders, the value of 1.93 was not been used. Similarly for cigarettes is $exp(-0.20 + 0.22) = 1.02$ and for marijuana is $exp(-0.20) = 0.82$.

d) The odds for a female to use alcohol is $exp(1.93 + 0.37) = 9.97$ times that for marijuana. Now, because we are comparing the use of substances we do not take into account the value of -0.20 which is solely for gender. Comparing cigarettes and marijuana the odds for a female to use the former is $exp(0.86 + 0.22) = 2.94$ that of the latter.

e) The odds for males to use alcohol is $exp(1.93) = 6.89$ times the odds of using marijuana. Note that the the interaction term cancels out. Comparing cigarettes and marijuana, the odds for male to use the former are $exp(0.86) = 2.36$ times that for the latter.

The interaction can be interpreted as that the females are more likely than males to use a given substance, except for marijuana

Problem 4

Since we have already constructed the data frame (`tb9.1b`) we just need to add a second time variable with the alternate coding and order the grouping variable

```
> tb9.1b <- read.table('supp_data/tb9-1', header = TRUE)
> tb9.1b$time2 <- ifelse(tb9.1b$time == 1, 2, ifelse(tb9.1b$time ==
+     0, 1, 4))
> tb9.1b <- tb9.1b[order(tb9.1b$case), ]
```

Then we can fit the models (Note that the one with the original coding was already fitted)

```
> fit.gee4 <- gee(outcome ~ diagnose + treat * time, id = case,
+     family = binomial, corstr = "exchangeable", data = tb9.1b)
```

```
(Intercept)    diagnose       treat        time  treat:time
-0.02798843 -1.31391092 -0.05960381  0.48241209  1.01744498
```

```
> fit2.gee4 <- gee(outcome ~ diagnose + treat * time2, id = case,
+     family = binomial, corstr = "exchangeable", data = tb9.1b)
```

```
(Intercept)    diagnose       treat       time2 treat:time2
 -0.2924555  -1.3022146  -0.6420318   0.3179815   0.7080492
```

With the different codings the effect of the initial diagnose is practically the same, a negative effect on the log odds of a normal response, however the coding does affect the treatment effect both for the first week of observation (`treatnew`) and the following weeks (`treatnew:time`). Interesting, for the second coding, the log odds of a normal response are lower with the new treat at the first ($exp(-0.64) = 0.53$)

Table 2: GEE model fitting for the depression data (table 9.1). Two codings were used for the time variable: Coding 1 = 0, 1, 2 and Coding 2 = 1, 2, 4

	Estimate	
	Coding 1	Coding 2
(Intercept)	-0.03	-0.29
diagnosesevere	-1.31	-1.30
treatnew	-0.06	-0.64
time	0.48	0.32
treatnew:time	1.02	0.71

and second ($exp(-0.64 + 0.71 \times 2) = 2.18$) observation, this compared to the first coding (0.94 and 2.61), however at the last observation the situation reverses, 9.03 for the second coding and 7.24 for the first one.
These differences are discussed in page 278 of Agresti when the time reflects cumulative dose effect. Nevertheless note that the coding doesn't change the fact that the new drug improves the state of the patients who took it.

Problem 5

As it's usual for GEE models we need to construct a data frame where each row represents a subject and collapse the observations for all the years into a single column (`reshape` function)

```
> tb9.8 <- data.frame(expand.grid(Y10=c(0, 1), Y9=c(0, 1), Y8=c(0, 1),
+                                 Y7=c(0, 1), smoking=c(0, 1)),
+                     count=c(237, 10, 15, 4, 16, 2, 7, 3, 24, 3, 3, 2, 6, 2, 5, 11,
+                             118, 6, 8, 2, 11, 1, 6, 4, 7, 3, 3, 1, 4, 2, 4, 7))
> tb9.8long <- reshape(tb9.8, varying = c("Y7", "Y8", "Y9", "Y10"),
+     direction = "long", timevar = "year", v.names = "outcome",
+     times = seq(7, 10))
> tb9.8long <- tb9.8long[rep(1:nrow(tb9.8long), tb9.8long$count),
+     ]
> tb9.8long$case <- 1:537
> tb9.8long <- tb9.8long[order(tb9.8long$case), ]
> row.names(tb9.8long) <- 1:nrow(tb9.8long)
> tb9.8long <- within(tb9.8long, {
+     rm(id, count)
+ })
```

Then we can fit the GEE model and compare with the Markov chain model already fitted in the notes corresponding to this chapter (`fit9.8`).

```
> fit.gee5 <- gee(outcome ~ smoking + year, id = case, family = binomial,
+     corstr = "exchangeable", data = tb9.8long)
```

```
(Intercept)     smoking        year
 -0.8630198   0.2721386  -0.1134128
```

Table 3: Estimates for the Child's respiratory illness from table 9.8 obtained with GEE and a first order Markov chain

	GEE	Markov chain
Intercept	-0.863	-0.293
Smoking	0.272	0.296
Time	-0.113	-0.243
Previous	–	2.211

This is summarized in the following table. Both the `time` and `smoking` effects are weak in the GEE model compared with the Markov chain model, even though, as it's mentioned in Agresti, the inclusion of a previous outcome weakens the effect of the other predictors. Despite the differences in magnitude the direction of them are the same, *i.e.* if the mother had smoked during the first year the odds for a child to have a respiratory illness increase (31% and 34% higher than those children whose mother had not smoked); regarding the time the more time passes the less likely is to have the illness.

Problem 6

First we create a data frame that has in one column the observations for all the treatments (`reshape` function). Then we repeat all the rows `count` times in order to create an ungrouped data frame. Latter we can remove the unnecessary variables.

```
> tb9.9 <- read.table('supp_data/tb9-9', header = TRUE)
> tb9.9b <- reshape(tb9.9, varying = c("A", "B", "C"), direction = "long",
+     v.names = "outcome", timevar = "treat", times = LETTERS[1:3])
> tb9.9b$treat <- factor(tb9.9b$treat, levels = LETTERS[1:3])
> tb9.9b <- tb9.9b[rep(1:nrow(tb9.9b), tb9.9b$count), ]
> tb9.9b <- within(tb9.9b, {
+     rm(count, id)
+ })
> row.names(tb9.9b) <- 1:nrow(tb9.9b)
```

Now, regarding the grouping variable (I think) because it is mentioned that subjects are nested within each sequence, we need to set the variable differently, grouping six sequence that go from 1 to each row marginal.

```
> tb9.9b$case <- c(1:15, 1:16, 1:15, 1:12, 1:14, 1:14)
> tb9.9b <- tb9.9b[order(tb9.9b$case), ]
```

Then we can fit the model

```
> summary(gee(outcome ~ -1 + seq + treat, id = case, family = binomial,
+     corstr = "exchangeable", data = tb9.9b))$coefficients
```

```
    seqABC     seqACB     seqBAC     seqBCA     seqCAB     seqCBA
-1.0314607 -0.7733839 -0.9078804 -0.9700100 -1.3133282 -1.5610477
    treatB     treatC
 1.9917375  2.5078182
         Estimate Naive S.E.   Naive z Robust S.E.  Robust z
seqABC -1.0159475  0.4394912 -2.311644   0.5266555 -1.929055
```

```
seqACB -0.7968132  0.4299927 -1.853085   0.5726704 -1.391399
seqBAC -0.8922848  0.4400726 -2.027585   0.4235151 -2.106855
seqBCA -0.7424225  0.4781889 -1.552572   0.5067194 -1.465155
seqCAB -1.2152791  0.4497817 -2.701931   0.6394355 -1.900550
seqCBA -1.4643945  0.4506852 -3.249262   0.6264402 -2.337644
treatB  1.9858731  0.3306172  6.006563   0.7067155  2.810004
treatC  2.5095193  0.3573354  7.022868   0.7058672  3.555229
```

a) The estimates can be interpreted as, that the odds for relieve by taking a low drug-dose are $exp(1.99) = 7.32$ times the odds for relieve by taking the placebo, also the odds of relieve by taking a high drug-dose are $exp(2.51) = 12.3$ times that for placebo.

b) This question ask about the combination that yields the highest value for $logit[P(Y_{i(k)t} = 1)] = \alpha_k + beta_t$. Based on the model estimates, the highest value is obtained for the combination of a placebo (A), then a high drug-dose (C) and a low drug-dose (B).

Note about the grouping variable I'm not sure if the logic used for creating the `case` variable was the correct one, however it's curious that if we set instead the sequence as the grouping variable the estimates are the same but an error message is reported.

Problem 7

a) According to Agresti (2007, p. 370), because farmers can select any number of sources, a given response may vary from zero to five and multinomial distributions does not apply to the cells (40, resulting from the combination of 2 education levels, 4 farm sizes and 5 sources of information).

b) First let's describe the model used.

$$logit[P(Y_{ist} = 1)] = \alpha_t + \beta_t s$$

Where t represents the source of information (1=Professional consultant, 2=Veterinarian, 3=State or local extension service, 4=Magazines, 5=Feed companies and reps.) and s represents the size of the farm in number of pigs (1=less than a thousand, 2=between one and two thousand, 3=between two and five thousand, 4=more than five thousand).
In this model I don't understand the meaning of the sub index for the intercept (α_t). But before the model fitting first take look at the data construction.

```
> tb9.10 <- data.frame(expand.grid(sourc.D = c(1, 0), sourc.C = c(1,
+     0), sourc.B = c(1, 0), sourc.A = c(1, 0), sourc.E = c(1,
+     0), pigs = c("<1", "1-2", "2-5", ">5"), edu = c(0, 1)), count = c(1,
+     0, 0, 0, 0, 0, 0, 0, 2, 1, 1, 2, 1, 1, 5, 3, 0, 0, 0, 0,
+     0, 0, 0, 1, 1, 0, 0, 5, 4, 7, 7, 0, 2, 0, 0, 0, 0, 0, 0,
+     0, 4, 0, 0, 4, 1, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
+     0, 5, 0, 3, 4, 0, 3, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 1, 2,
+     0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 2, 0, 1, 4, 0,
+     2, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 2, 1, 0, 0,
+     2, 1, 0, 1, 6, 0, 1, 1, 1, 0, 0, 6, 0, 3, 0, 0, 0, 0, 0,
+     0, 0, 4, 0, 1, 1, 0, 0, 2, 11, 0, 0, 0, 0, 0, 0, 0, 0, 4,
+     0, 1, 2, 4, 6, 14, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 1,
+     0, 0, 1, 6, 0, 0, 0, 0, 1, 0, 0, 1, 2, 1, 0, 4, 2, 7, 14,
```

```
+     0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 3, 1, 0,
+     0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 4, 4, 0, 1, 0, 0, 0, 0,
+     0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 2, 1, 1, 0, 0, 0, 1, 0, 10,
+     0, 0, 0, 4, 1, 2, 4, 0))
> colnames(tb9.10) <- c("D", "C", "B", "A", "E", "pigs", "edu",
+     "count")
> tb9.10long <- reshape(tb9.10, varying = c("A", "B", "C", "D",
+     "E"), direction = "long", v.names = "outcome", timevar = "source",
+     times = LETTERS[1:5])
> tb9.10long <- tb9.10long[rep(1:nrow(tb9.10long), tb9.10long$count),
+     ]
> rownames(tb9.10long) <- 1:nrow(tb9.10long)
> tb9.10long <- within(tb9.10long, {
+     rm(id, count)
+ })
> tb9.10long$sourc <- factor(tb9.10long$sourc, levels = LETTERS[5:1])
> tb9.10long$pigs3 <- ifelse(tb9.10long$pigs == "<1", 1, ifelse(tb9.10long$pigs ==
+     "1-2", 2, ifelse(tb9.10long$pigs == "2-5", 3, 4)))
> tb9.10long$case <- 1:262
> tb9.10long <- tb9.10long[order(tb9.10long$case), ]
```

Note, as in the other examples that the data frame must be ordered according to the `case` vector.

Now, as it was said above, to match the values reported in table 9.11 we need to fit the following model:

```
> fit.gee7 <- gee(outcome ~ -1 + sourc + sourc:pigs3, id = case,
+     corstr = "exchangeable", family = binomial, data = tb9.10long)
```

```
      sourcE       sourcD       sourcC       sourcB       sourcA
 -0.06498563   0.50338932  -0.13669470  -0.81079309  -4.45313522
sourcE:pigs3 sourcD:pigs3 sourcC:pigs3 sourcB:pigs3 sourcA:pigs3
 -0.24462408  -0.22648204  -0.19533240   0.07281225   1.06654271
```

which is a model with no intercept (`-1`) and where the effect of the size of the farm according to the source of information is represented by `sourc:pigs3`.

The relationship among the size of the farm across the different sources of information can be illustrated by constructing a set of tables that classify the proportion of yes and no responses for the different sources of information and different sizes of farm. This is done with the `xtabs` function wrapped in a loop,

```
> for (i in 1:4) {
+     print(round(xtabs(~outcome + sourc + pigs, data = tb9.10long)[,
+         , i]/margin.table(xtabs(~outcome + sourc + pigs, data = tb9.10long)[,
+         , i], 2)[1], 2))
+ }
```

```
       sourc
outcome    E    D    C    B    A
```

```
      0 0.59 0.42 0.59 0.69 0.95
      1 0.41 0.58 0.41 0.31 0.05
       sourc
outcome    E    D    C    B    A
      0 0.64 0.52 0.64 0.64 0.94
      1 0.36 0.48 0.36 0.36 0.06
       sourc
outcome    E    D    C    B    A
      0 0.60 0.50 0.60 0.60 0.81
      1 0.40 0.50 0.40 0.40 0.19
       sourc
outcome    E    D    C    B    A
      0 0.79 0.61 0.75 0.66 0.52
      1 0.21 0.39 0.25 0.34 0.48
```

Note that the proportion of yes responses for source A increases steadily with the size of the farm, particularly in the last category. The other source that has a significant value is source D, that decreases but not as steadily as source A and also the magnitude of the difference isn't that big. For the rest of the sources the differences are less clear.

Problem 8

First we construct the data frame

```
> tb10.4 <- read.table('supp_data/tb10-4', header=TRUE)
> tb10.4long <- reshape(tb10.4, direction = "long", varying = c("Q1",
+     "Q2", "Q3"), v.names = "response", timevar = "question")
> tb10.4long <- tb10.4long[rep(1:nrow(tb10.4long), tb10.4long$count),
+     ]
> tb10.4long <- within(tb10.4long, {
+     rm(count, id)
+ })
> tb10.4long$case <- 1:1850
> tb10.4long <- tb10.4long[order(tb10.4long$case), ]
> tb10.4long$question <- factor(tb10.4long$question, levels = 3:1)
> tb10.4long$gender <- factor(tb10.4long$gender, levels = c("M",
+     "F"))
```

a) Remember that the *unstructured* correlation matrix permits variation among pairs, in this case among pairs of responses.

```
> summary(gee(response ~ question + gender, id = case, corstr = "unstructured",
+     family = "binomial", data = tb10.4long))$working.correlation
```

```
 (Intercept)    question2    question1      genderF
-0.125407576  0.052017989  0.149347113  0.003582051
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.8248498 0.7958825
[2,] 0.8248498 1.0000000 0.8312594
[3,] 0.7958825 0.8312594 1.0000000
```

The values are the same as those reported in Agresti. Now regarding the question, first we have to define what do we interpret the term *"reasonable"*. Section 10.2.4 Agresti depicts 3 different scenarios for abortion (1 = if the family has a very low income and cannot afford any more children, 2 = when the woman is not married and does not want to marry the man, and 3 = when the woman wants it for any reason) where the resulting pairs may have a different response from those interviewed. In this sense a *"reasonable"* working correlation matrix could be the *unstructured*. However looking at the values obtained with this one we see that they are very similar so an *exchangeable* correlation matrix (that assumes the same value for all pairs) could work as well. In fact Agresti (2007, p. 281) recommends the latter one as an starting point and when we don't have a clear idea about the correlation matrix. Also important is the remark in page 306 about the position of Americans about abortion. Agresti indicates that Americans "tend to be either uniformly opposed to legalized abortion, regardless of the circumstances, or uniformly in favor of it" which explains the similar values.

b) Same values as those reported in Agresti are found with the command,

```
> summary(gee(response ~ question + gender , id = case ,
+                corstr ="exchangeable" , family ="binomial" ,
+                data = tb10.4long))$coefficients
```

```
 (Intercept)    question2    question1      genderF
-0.125407576  0.052017989  0.149347113  0.003582051
                Estimate Naive S.E.     Naive z Robust S.E.
(Intercept) -0.125325730 0.06782579 -1.84775925  0.06758212
question2    0.052017986 0.02815145  1.84779075  0.02704703
question1    0.149347107 0.02814374  5.30658404  0.02973865
genderF      0.003437873 0.08790630  0.03910838  0.08784072
               Robust z
(Intercept) -1.85442135
question2    1.92324179
question1    5.02198729
genderF      0.03913758
```

These can be interpreted as that the odds for a female to be in favor of abortion are $exp(0.005) = 1.01$ times those of males; in other words the odds for females are less than 1% higher than that for males, this controlling on the question type. Regarding the question and controlling for gender, the odds of being in favor of abortion for question 1 instead of question 3 is $exp(0.149) = 1.16$ times or 16% higher, while the odds for question 2 instead of 3 are $exp(0.052) = 1.05$ times or 5% higher. Again, this shows a more or less uniform response, though question 1 seems a bit more likely to obtain support.

Problem 9

First we construct the data.

```
> tb10.8long <- read.table("supp_data/tb10-8long", header = TRUE)
> tb10.8long <- tb10.8long[order(tb10.8long$study), ]
```

a) We can assume that because of unmeasured factors such as doctor's experience or environment quality the success of the treatment could vary across centers (studies) so we fit a model with an unstructured correlation matrix, however the model fails to fit the model. Now with an exchangeable correlation matrix the results are the following,

```
> summary(gee(response~treat, id=study, family="binomial", corstr="exchangeable", data=tb10.8lor
```

```
(Intercept)       treat
 -0.4571371  -1.0082946
             Estimate Naive S.E.   Naive z Robust S.E.  Robust z
(Intercept) -0.316674  0.1052226 -3.009562   0.1773677 -1.785410
treat       -1.023080  0.1023785 -9.993113   0.2173890 -4.706219
```

b) A way to compare the two surgeries is testing the null hypothesis $H_0 : \beta_{treat} = 0$. Because GEEs can not use Likelihood Ratio approximations we are left only with large sample Wald statistics. According to the results from the GEE model the estimate and its standard error are, respectively -1.023080 and 0.2173890, thus the z-value is $-1.023080/0.2173890 = -4.706219$. Now remember that the square of this value has an approximate chi-square deviation with 1 degree of freedom, so the p-value for the null hypothesis is 0 which indicates that there is difference between the surgeries. The estimate has a 95% confidence interval of:

```
> -1.02308 + c(-1, 1) * qnorm(0.95) * sqrt(0.217389)
```

```
[1] -1.7899929 -0.2561671
```

Thus we can be 95% sure that the odds of an adverse event are at least 33% lower for then new surgery and at most 84% lower.

c) First we fit the GLM model.

```
> summary(glm(response ~ treat + study, family = binomial, data = tb10.8long))$coefficients
```

```
               Estimate  Std. Error   z value     Pr(>|z|)
(Intercept) -0.424470311 0.118826352 -3.572190 3.540084e-04
treat       -1.007740754 0.109010693 -9.244421 2.365174e-20
study       -0.001534218 0.004577706 -0.335150 7.375119e-01
```

The estimate for the treatment effect is similar to that obtained with the GEE (-1.01 vs -1.02). This is not surprise since the correlation coefficient for the GEE's clusters is very low (0.07).

Problem 10

First we construct the data. For this we start by transforming the data frame, previously constructed in chapter 8 (`tb8.14`).

```
> tb8.14 <- read.table('supp_data/tb8-14', header = TRUE)
> tb8.14long <- reshape(tb8.14, direction = "long", varying = c("Pre.sex",
+     "Post.sex"), v.names = "outcome")
> tb8.14long <- within(tb8.14long, {
+     rm(id, symm, score)
+ })
> tb8.14long <- tb8.14long[rep(1:nrow(tb8.14long), tb8.14long$C),
+     ]
> rownames(tb8.14long) <- 1:nrow(tb8.14long)
```

```
> tb8.14long$case <- 1:475
> tb8.14long <- tb8.14long[order(tb8.14long$case, tb8.14long$time),
+     ]
> tb8.14long$outcome <- ordered(tb8.14long$outcome, levels = 1:4)
```

Two important things to note, first that as with previous models, the grouping variable as well as the time variable are ordered, second, the outcome variable is an `ordered` vector.
Then we can fit the model with the `ordgee` function

```
> summary(ordgee(outcome ~ time, id = case, corstr = "independence",
+     rev = TRUE, control = geese.control(maxit = 100), data = tb8.14long))
```

```
Call:
ordgee(formula = outcome ~ time, id = case, data = tb8.14long,
    corstr = "independence", control = geese.control(maxit = 100),
    rev = TRUE)

Mean Model:
 Mean Link:                 logit
 Variance to Mean Relation: binomial

 Coefficients:
         estimate    san.se      wald            p
Inter:1 -4.421038 0.3662038 145.74811 0.000000e+00
Inter:2 -3.855679 0.3604677 114.41129 0.000000e+00
Inter:3 -2.745549 0.3465845  62.75384 2.331468e-15
time     3.452183 0.3168031 118.74321 0.000000e+00

Scale is fixed.

Correlation Model:
 Correlation Structure:     independence

Returned Error Value:    0
Number of clusters:   475   Maximum cluster size: 2
```

The interpretation of the time estimate is that, the odds of judging extra-marital sex in the lower end is $exp(3.45) = 31.5$ times the odds of judging the pre-marital sex in the same category. The interpretation is a bit different than that found in exercise 8.15, where the estimate referred to the inverse odds ratio, that is the odds of judging pre-marital sex at a lower value of the scale instead that that of extra-marital sex. However the conclusion is the same, extra-marital sex is judged as a more wrong attitude than pre-marital sex.

Problem 11

First we construct the data frame

```
> tb7.25 <- read.table('supp_data/tb7-25', header = TRUE)
> tb7.25long <- reshape(tb7.25, direction = "long", varying = c("E",
+     "H", "C", "L"), v.names = "outcome", timevar = "question",
```

```
+     times = c("env", "health", "cities", "law"))
> tb7.25long <- tb7.25long[rep(1:nrow(tb7.25long), tb7.25long$N),
+     ]
> rownames(tb7.25long) <- 1:nrow(tb7.25long)
> tb7.25long <- within(tb7.25long, {
+     rm(N, id)
+ })
> tb7.25long$question <- factor(tb7.25long$question, levels = c("env",
+     "health", "cities", "law"))
> tb7.25long$case <- 1:607
> tb7.25long <- tb7.25long[order(tb7.25long$case), ]
> tb7.25long$outcome <- ordered(tb7.25long$outcome)
```

Then we fit the model.

```
> fit.ordgee.11 <- ordgee(outcome ~ question, id = case, data = tb7.25long,
+     corstr = "independence", rev = TRUE, control = geese.control(maxit = 100))
> fit2.ordgee.11 <- ordgee(outcome ~ question, id = case, data = tb7.25long,
+     corstr = "exchangeable", rev = TRUE, control = geese.control(maxit = 100))
```

Note that in this case we can select the exchangeable beside the independence correlation matrix beside, however the estimates are practically the same.

Table 4: GEE estimates for the data in table 7.25 with two correlation matrices

	Correlation matrix			
	Exchangeable		Independence	
	estimate	san.se	estimate	san.se
Inter:1	0.93065417	0.1310183	0.93758531	0.1299585
Inter:2	2.81535458	0.1542325	2.81259565	0.1549396
questionhealth	-0.01908055	0.1848736	-0.01857888	0.1857996
questioncities	-2.26777822	0.1431653	-2.26117481	0.1482185
questionlaw	-0.21553857	0.1855937	-0.23695831	0.1862137

This estimates can be interpreted as the odds of thinking that government spending is lower for any category, compared with that for environment; for example the odds of thinking that the government spending is lower for health than environment is $exp(-0.0191) = 0.98$ or 2% lower than the reverse *i.e.* the odds of thinking that government's spending on the environment is lower than that for health. Note that because all the estimates are negative, the odds of thinking that in general people think that government spending on environment is lower than any of the other categories, except maybe health. In other words in people's opinion environment and health are priorities and cities and law enforcement are not so much.

Problem 12

This question asks for a first order Markov–chain model, so first we construct the data.

```
> tb9.6 <- data.frame(expand.grid(t2 = c(10, 25, 45, 75), t1 = c(10,
+     25, 45, 75), treat = c(0, 1)), count = c(7, 4, 2, 1, 14,
+     5, 1, 0, 6, 9, 18, 2, 4, 11, 14, 22, 7, 4, 1, 0, 11, 5, 2,
```

```
+     2, 13, 23, 3, 1, 9, 17, 13, 8))
> tb9.6long <- tb9.6[rep(1:nrow(tb9.6), tb9.6$count), ]
> rownames(tb9.6long) <- 1:nrow(tb9.6long)
> tb9.6long$t1f <- factor(tb9.6long$t1, levels = c(10, 25, 45,
+     75))
> tb9.6long$t2 <- ordered(tb9.6long$t2, levels = c(10, 25, 45,
+     75))
> tb9.6long$t1 <- -tb9.6long$t1
> tb9.6long$treat <- -tb9.6long$treat
```

Then we fit the model with the `polr` function from the `MASS` package. Now, because we are using this function we changed the sign of the explanatory variables in the last part of the above commands (see Agresti 2007, p. 121).

```
> polr(t2 ~ treat * t1, data = tb9.6long)
```

```
Call:
polr(formula = t2 ~ treat * t1, data = tb9.6long)

Coefficients:
      treat          t1    treat:t1
-0.21726202 -0.05283874 -0.02174072

Intercepts:
   10|25    25|45    45|75
1.107579 2.810701 4.326973

Residual Deviance: 572.1139
AIC: 584.1139
```

a) Because we change the sing of the explanatory variables, when we calculate the log odds ratio for the two initial times we also have to use values with their sing changed.

```
[1] 0.4346692
```

```
[1] 1.847816
```

This values can be interpreted as: the odds ratio of having a response at least as equal to the first one under the treatment is 1.54 times when the first response was on the lowest scale (*i.e.* the individual fell asleep within less than 20 minutes) and is 6.35 times when the first response was on the highest scale (*i.e.* the individual fell asleep within more than 50 minutes).

b) Removing the interaction term and changing the initial response from a quantitative variable to a factor we get.

```
> round(summary(polr(t2 ~ treat + t1f, data = tb9.6long))$coefficients,
+     3)
```

```
       Value Std. Error t value
treat  0.911      0.249   3.664
t1f25 -0.366      0.508  -0.720
t1f45  1.154      0.441   2.616
t1f75  2.307      0.449   5.141
10|25 -0.306      0.405  -0.755
25|45  1.425      0.419   3.402
45|75  2.905      0.445   6.524
```

For the treatment effect the odds of having a response at least equal to the first one is $exp(0.911) = 2.49$ times than that without the treatment. Controlling for the treatment, the odds of having a response at least equal to the first one is $exp(-0.366) = 0.69$ times than when the first response was between 20 and 30 minutes and $exp(1.154) = 3.17$ times than when the first response was between 30 and 60 minutes.

c) Adding an interaction term to the previous model.

```
> round(summary(polr(t2 ~ treat * t1f, data = tb9.6long))$coefficients,
+     3)
```

```
             Value Std. Error t value
treat        0.527      0.769   0.686
t1f25       -1.022      0.710  -1.439
t1f45        1.557      0.610   2.553
t1f75        2.724      0.609   4.476
treat:t1f25 -1.372      1.012  -1.356
treat:t1f45  0.775      0.873   0.888
treat:t1f75  0.779      0.861   0.905
10|25       -0.147      0.524  -0.280
25|45        1.635      0.540   3.028
45|75        3.164      0.568   5.572
```

As Agresti notes the estimates suggest that the treatment is more effective for the highest initial responses. This isn't unusual since it would be difficult to match an already low response.

Problem 13

As it's explained in Thompson and summarized in the notes (page 9) we need to construct a table with a column for the smoking status, the years evaluated and two columns for the previous responses; now because of the latter the years evaluated get reduced to only years 9 and 10.

```
> tb9.8c <- data.frame(expand.grid(prev2 = c(0, 1), prev1 = c(0,
+     1), t = 9:10, smoking = c(0, 1)))
> tb9.8c$yes <- c(sum(tb9.8[tb9.8$Y9 == 1 & tb9.8$smoking == 0 &
+     tb9.8$Y8 == 0 & tb9.8$Y7 == 0, "count"]), sum(tb9.8[tb9.8$Y9 ==
+     1 & tb9.8$smoking == 0 & tb9.8$Y8 == 0 & tb9.8$Y7 == 1, "count"]),
+     sum(tb9.8[tb9.8$Y9 == 1 & tb9.8$smoking == 0 & tb9.8$Y8 ==
+         1 & tb9.8$Y7 == 0, "count"]), sum(tb9.8[tb9.8$Y9 == 1 &
+         tb9.8$smoking == 0 & tb9.8$Y8 == 1 & tb9.8$Y7 == 1, "count"]),
+     sum(tb9.8[tb9.8$Y10 == 1 & tb9.8$smoking == 0 & tb9.8$Y9 ==
+         0 & tb9.8$Y8 == 0, "count"]), sum(tb9.8[tb9.8$Y10 ==
```

```
+         1 & tb9.8$smoking == 0 & tb9.8$Y9 == 0 & tb9.8$Y8 ==
+         1, "count"]), sum(tb9.8[tb9.8$Y10 == 1 & tb9.8$smoking ==
+         0 & tb9.8$Y9 == 1 & tb9.8$Y8 == 0, "count"]), sum(tb9.8[tb9.8$Y10 ==
+         1 & tb9.8$smoking == 0 & tb9.8$Y9 == 1 & tb9.8$Y8 ==
+         1, "count"]), sum(tb9.8[tb9.8$Y9 == 1 & tb9.8$smoking ==
+         1 & tb9.8$Y8 == 0 & tb9.8$Y7 == 0, "count"]), sum(tb9.8[tb9.8$Y9 ==
+         1 & tb9.8$smoking == 1 & tb9.8$Y8 == 0 & tb9.8$Y7 ==
+         1, "count"]), sum(tb9.8[tb9.8$Y9 == 1 & tb9.8$smoking ==
+         1 & tb9.8$Y8 == 1 & tb9.8$Y7 == 0, "count"]), sum(tb9.8[tb9.8$Y9 ==
+         1 & tb9.8$smoking == 1 & tb9.8$Y8 == 1 & tb9.8$Y7 ==
+         1, "count"]), sum(tb9.8[tb9.8$Y10 == 1 & tb9.8$smoking ==
+         1 & tb9.8$Y9 == 0 & tb9.8$Y8 == 0, "count"]), sum(tb9.8[tb9.8$Y10 ==
+         1 & tb9.8$smoking == 1 & tb9.8$Y9 == 0 & tb9.8$Y8 ==
+         1, "count"]), sum(tb9.8[tb9.8$Y10 == 1 & tb9.8$smoking ==
+         1 & tb9.8$Y9 == 1 & tb9.8$Y8 == 0, "count"]), sum(tb9.8[tb9.8$Y10 ==
+         1 & tb9.8$smoking == 1 & tb9.8$Y9 == 1 & tb9.8$Y8 ==
+         1, "count"]))
> tb9.8c$no <- c(sum(tb9.8[tb9.8$Y9 == 0 & tb9.8$smoking == 0 &
+     tb9.8$Y8 == 0 & tb9.8$Y7 == 0, "count"]), sum(tb9.8[tb9.8$Y9 ==
+     0 & tb9.8$smoking == 0 & tb9.8$Y8 == 0 & tb9.8$Y7 == 1, "count"]),
+     sum(tb9.8[tb9.8$Y9 == 0 & tb9.8$smoking == 0 & tb9.8$Y8 ==
+         1 & tb9.8$Y7 == 0, "count"]), sum(tb9.8[tb9.8$Y9 == 0 &
+         tb9.8$smoking == 0 & tb9.8$Y8 == 1 & tb9.8$Y7 == 1, "count"]),
+     sum(tb9.8[tb9.8$Y10 == 0 & tb9.8$smoking == 0 & tb9.8$Y9 ==
+         0 & tb9.8$Y8 == 0, "count"]), sum(tb9.8[tb9.8$Y10 ==
+         0 & tb9.8$smoking == 0 & tb9.8$Y9 == 0 & tb9.8$Y8 ==
+         1, "count"]), sum(tb9.8[tb9.8$Y10 == 0 & tb9.8$smoking ==
+         0 & tb9.8$Y9 == 1 & tb9.8$Y8 == 0, "count"]), sum(tb9.8[tb9.8$Y10 ==
+         0 & tb9.8$smoking == 0 & tb9.8$Y9 == 1 & tb9.8$Y8 ==
+         1, "count"]), sum(tb9.8[tb9.8$Y9 == 0 & tb9.8$smoking ==
+         1 & tb9.8$Y8 == 0 & tb9.8$Y7 == 0, "count"]), sum(tb9.8[tb9.8$Y9 ==
+         0 & tb9.8$smoking == 1 & tb9.8$Y8 == 0 & tb9.8$Y7 ==
+         1, "count"]), sum(tb9.8[tb9.8$Y9 == 0 & tb9.8$smoking ==
+         1 & tb9.8$Y8 == 1 & tb9.8$Y7 == 0, "count"]), sum(tb9.8[tb9.8$Y9 ==
+         0 & tb9.8$smoking == 1 & tb9.8$Y8 == 1 & tb9.8$Y7 ==
+         1, "count"]), sum(tb9.8[tb9.8$Y10 == 0 & tb9.8$smoking ==
+         1 & tb9.8$Y9 == 0 & tb9.8$Y8 == 0, "count"]), sum(tb9.8[tb9.8$Y10 ==
+         0 & tb9.8$smoking == 1 & tb9.8$Y9 == 0 & tb9.8$Y8 ==
+         1, "count"]), sum(tb9.8[tb9.8$Y10 == 0 & tb9.8$smoking ==
+         1 & tb9.8$Y9 == 1 & tb9.8$Y8 == 0, "count"]), sum(tb9.8[tb9.8$Y10 ==
+         0 & tb9.8$smoking == 1 & tb9.8$Y9 == 1 & tb9.8$Y8 ==
+         1, "count"]))
> tb9.8c <- within(tb9.8c, {
+     total <- yes + no
+     prop <- yes/total
+     rm(no)
+ })
```

Note that the data frame contains 16 rows corresponding to the combination of 2 smoking status, 2

years evaluated, 2 observations for year $t-1$ and 2 observations for year $t-2$[1]. Also note that two other vectors were created, one for the proportion of positive outcomes and other for the total number of outcomes, they are used in the model fitting.
Then we can fit the model with the regular function to fit Generalized Linear Models, `glm`.

```
> summary(fit9.8a <- glm(prop ~ prev1 + prev2 + t + smoking, data = tb9.8c,
+     family = binomial, weight = total))$coefficients
```

```
             Estimate Std. Error    z value     Pr(>|z|)
(Intercept)  1.3736439  1.9006307  0.7227305 4.698454e-01
prev1        1.9451576  0.2151795  9.0396971 1.571065e-19
prev2        1.1480945  0.2231128  5.1458031 2.663788e-07
t           -0.4365775  0.2017036 -2.1644511 3.042975e-02
smoking      0.1740876  0.2044236  0.8516023 3.944349e-01
```

Adding a second "previous" observation does add predictive power, although lower than the most recent observation, but still higher than the rest of the predictors.

b) The magnitude of the smoking effect reduces from 0.296 to 0.174 but is still significant.

Problem 14

Just like the other problems that ask for a Markov–chain model we need to construct two data frames, the first with the original data, in this case with the counts for all the possible observations at the three evaluations, two treatments and two initial diagnoses.

```
> tb9.1c <- data.frame(expand.grid(t3 = c("N", "A"), t2 = c("N",
+     "A"), t1 = c("N", "A"), treat = c(0, 1), diagnose = c("mild",
+     "severe")), count = c(16, 13, 9, 3, 14, 4, 15, 6, 31, 0,
+     6, 0, 22, 2, 9, 0, 2, 2, 8, 9, 9, 15, 27, 28, 7, 2, 5, 2,
+     31, 5, 32, 6))
> tb9.1c$T <- paste(tb9.1c$t1, tb9.1c$t2, tb9.1c$t3, sep = "")
> xtabs(count ~ treat + T + diagnose, data = tb9.1c)
```

```
, , diagnose = mild

     T
treat AAA AAN ANA ANN NAA NAN NNA NNN
    0   6  15   4  14   3   9  13  16
    1   0   9   2  22   0   6   0  31

, , diagnose = severe

     T
treat AAA AAN ANA ANN NAA NAN NNA NNN
    0  28  27  15   9   9   8   2   2
    1   6  32   5  31   2   5   2   7
```

[1]During the construction of the table a mistake in the number of elements in the yes and no column caused a problem in the fitting, thus this step must be done carefully.

The last two commands where to verify if the data frame was constructed correctly. Note that the values are similar to those of table 9.1 in Agresti, though with the possible outcomes reversed.
Then we can construct the second data frame, one resulting from the combination of 2 previous outcomes, 2 evaluations, 2 initial diagnoses and 2 treatments, namely a data frame with 16 possible combinations/rows.

```
> tb9.1d <- data.frame(expand.grid(prev1 = c("N", "A"), t = 2:3,
+     diagnose = c("mild", "severe"), treat = c(0, 1)))
> tb9.1d$N <- c(sum(tb9.1c[tb9.1c$t2 == "N" & tb9.1c$treat == 0 &
+     tb9.1c$diagnose == "mild" & tb9.1c$t1 == "N", "count"]),
+     sum(tb9.1c[tb9.1c$t2 == "N" & tb9.1c$treat == 0 & tb9.1c$diagnose ==
+         "mild" & tb9.1c$t1 == "A", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "N" & tb9.1c$treat == 0 & tb9.1c$diagnose == "mild" &
+         tb9.1c$t2 == "N", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "N" & tb9.1c$treat == 0 & tb9.1c$diagnose == "mild" &
+         tb9.1c$t2 == "A", "count"]), sum(tb9.1c[tb9.1c$t2 ==
+         "N" & tb9.1c$treat == 0 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t1 == "N", "count"]), sum(tb9.1c[tb9.1c$t2 ==
+         "N" & tb9.1c$treat == 0 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t1 == "A", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "N" & tb9.1c$treat == 0 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t2 == "N", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "N" & tb9.1c$treat == 0 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t2 == "A", "count"]), sum(tb9.1c[tb9.1c$t2 ==
+         "N" & tb9.1c$treat == 1 & tb9.1c$diagnose == "mild" &
+         tb9.1c$t1 == "N", "count"]), sum(tb9.1c[tb9.1c$t2 ==
+         "N" & tb9.1c$treat == 1 & tb9.1c$diagnose == "mild" &
+         tb9.1c$t1 == "A", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "N" & tb9.1c$treat == 1 & tb9.1c$diagnose == "mild" &
+         tb9.1c$t2 == "N", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "N" & tb9.1c$treat == 1 & tb9.1c$diagnose == "mild" &
+         tb9.1c$t2 == "A", "count"]), sum(tb9.1c[tb9.1c$t2 ==
+         "N" & tb9.1c$treat == 1 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t1 == "N", "count"]), sum(tb9.1c[tb9.1c$t2 ==
+         "N" & tb9.1c$treat == 1 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t1 == "A", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "N" & tb9.1c$treat == 1 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t2 == "N", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "N" & tb9.1c$treat == 1 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t2 == "A", "count"]))
> tb9.1d$A <- c(sum(tb9.1c[tb9.1c$t2 == "A" & tb9.1c$treat == 0 &
+     tb9.1c$diagnose == "mild" & tb9.1c$t1 == "N", "count"]),
+     sum(tb9.1c[tb9.1c$t2 == "A" & tb9.1c$treat == 0 & tb9.1c$diagnose ==
+         "mild" & tb9.1c$t1 == "A", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "A" & tb9.1c$treat == 0 & tb9.1c$diagnose == "mild" &
+         tb9.1c$t2 == "N", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "A" & tb9.1c$treat == 0 & tb9.1c$diagnose == "mild" &
+         tb9.1c$t2 == "A", "count"]), sum(tb9.1c[tb9.1c$t2 ==
+         "A" & tb9.1c$treat == 0 & tb9.1c$diagnose == "severe" &
```

```
+         tb9.1c$t1 == "N", "count"]), sum(tb9.1c[tb9.1c$t2 ==
+         "A" & tb9.1c$treat == 0 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t1 == "A", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "A" & tb9.1c$treat == 0 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t2 == "N", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "A" & tb9.1c$treat == 0 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t2 == "A", "count"]), sum(tb9.1c[tb9.1c$t2 ==
+         "A" & tb9.1c$treat == 1 & tb9.1c$diagnose == "mild" &
+         tb9.1c$t1 == "N", "count"]), sum(tb9.1c[tb9.1c$t2 ==
+         "A" & tb9.1c$treat == 1 & tb9.1c$diagnose == "mild" &
+         tb9.1c$t1 == "A", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "A" & tb9.1c$treat == 1 & tb9.1c$diagnose == "mild" &
+         tb9.1c$t2 == "N", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "A" & tb9.1c$treat == 1 & tb9.1c$diagnose == "mild" &
+         tb9.1c$t2 == "A", "count"]), sum(tb9.1c[tb9.1c$t2 ==
+         "A" & tb9.1c$treat == 1 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t1 == "N", "count"]), sum(tb9.1c[tb9.1c$t2 ==
+         "A" & tb9.1c$treat == 1 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t1 == "A", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "A" & tb9.1c$treat == 1 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t2 == "N", "count"]), sum(tb9.1c[tb9.1c$t3 ==
+         "A" & tb9.1c$treat == 1 & tb9.1c$diagnose == "severe" &
+         tb9.1c$t2 == "A", "count"]))
> tb9.1d <- within(tb9.1d, {
+     total <- N + A
+     prop <- N/total
+     rm(A)
+ })
```

Then we can fit the model

```
> summary(glm(prop ~ diagnose + treat + t + prev1, weights = total,
+     data = tb9.1d, family = binomial))$coefficients
```

```
                 Estimate Std. Error    z value     Pr(>|z|)
(Intercept)    -1.7875157  0.4854507 -3.6821778 2.312501e-04
diagnosesevere -1.1616065  0.1915802 -6.0632922 1.333630e-09
treat           1.3591320  0.1822954  7.4556580 8.942040e-14
t               0.9870425  0.1801099  5.4802231 4.247898e-08
prev1A         -0.1531992  0.1895189 -0.8083584 4.188843e-01
```

All the estimates, though different in magnitude, have the same effect, for example the the odds of a normal response decrease with a "severe" initial diagnose (-1.162 in the Markov model vs -1.314 in the GEE model), but increases with the new treatment (1.359 in the Markov model vs 0.957[2] for the GEE model). Interesting the previous diagnose has no significant effect on the odds of a normal response. Finally, the time effect is higher in magnitude for the Markov model but that is probably because the GEE model has an interaction term. If we include an interaction term the values become even closer to those in the GEE model and the previous response still is non-significant.

[2]This value results from the equation, $logit[P(Y_t = 1)] = -0.028 - 1.314s - 0.060d + 0.482t + 1.017t \times d$, namely by the operation $-0.060 + 1.017$

Table 5: Estimates for the depression data (table 9.1) with a Markov model and a GEE

	Markov estimates		GEE estimates	
	Estimate	Std. Error	Estimate	Robust S.E.
(Intercept)	-0.255	0.376	-0.028	0.174
diagnosesevere	-1.178	0.191	-1.314	0.146
treat	-0.138	0.557	-0.059	0.229
t2	0.605	0.223	0.482	0.120
prev1A	-0.090	0.192	–	–
treat:t2	1.072	0.384	1.017	0.188

Problem 15

We could analyze the data either with a Markov chain model or with GEE, as it was pointed out by Agresti the former has the advantage of using LR based methods so we could test more accurately, and easily, the significance of a particular term in the model, however the inclusion of previous responses reduces the magnitude of the effects of the other predictors. In any case, before fitting any model it is useful to calculate the proportion of a particular outcome (in this case the proportion of obese children) through time. As it's shown in the notes (page 4) we do this by first creating a temporary variable with all the outcomes (positive and negative) and then dividing the positive outcomes by that variable. Worth noting is that both tables are constructed on a ungrouped data frame, which will also serve to fit a GEE model

```
> tb9.13 <- read.table('supp_data/tb9-13', header = TRUE)
> tb9.13long <- reshape(tb9.13, direction = "long", varying = c("resp.77",
+     "resp.79", "resp.81"), v.names = "outcome", times = seq(77,
+     81, 2))
> tb9.13long <- tb9.13long[rep(1:nrow(tb9.13long), tb9.13long$count),
+     ]
> tb9.13long <- within(tb9.13long, {
+     rm(id, count)
+ })
> tb9.13long$case <- 1:363
> temp <- xtabs(I(ifelse(tb9.13long$outcome == 0, 1, 1)) ~ gend +
+     time, data = tb9.13long)
> round(xtabs(outcome ~ gend + time, data = tb9.13long)/temp, 3)
```

```
    time
gend    77    79    81
   0 0.243 0.210 0.166
   1 0.159 0.203 0.181
```

Note that the proportion of obese children diminishes steadily for males, however for females first it increases but then decreases. It seems to be an interaction between gender and obesity.
Let's fit the interaction model first with a GEE. For this we need to make additional modifications, like order the data frame based on the `case` vector.

```
> tb9.13long <- tb9.13long[order(tb9.13long$case), ]
> tb9.13long$time2 <- ifelse(tb9.13long$time == 77, 1, ifelse(tb9.13long$time ==
+     79, 2, 3))
```

```
> rownames(tb9.13long) <- 1:nrow(tb9.13long)
> summary(geese(outcome ~ gend * time2, id = case, family = binomial,
+     corstr = "exchangeable", data = tb9.13long))[1:2]
```

```
$mean
              estimate     san.se      wald           p
(Intercept) -0.8830129 0.23372325 14.273494 0.0001580752
gend        -0.7736353 0.35128247  4.850199 0.0276429443
time2       -0.2377004 0.09717042  5.984000 0.0144362273
gend:time2   0.3117744 0.14035378  4.934381 0.0263274945

$correlation
       estimate     san.se     wald           p
alpha 0.5040955 0.07619733 43.76693 3.69903e-11
```

First of all, note that a different `time` variable was defined with the values, 1, 2 and 3. This was done because the original coding resulted in bigger estimates for the gender effect. Second, note that there is a significant correlation within clusters (0.5), because of this a model that treats all the observations as independent would yield biased estimates' standard errors.
Regarding the estimates themselves they correspond to the following model,

$$logit[P(Y_t = 1)] = \alpha + \beta_1 G + \beta_2 t + \beta_3 t \times G$$

Where $P(Y_t = 1)$ is the probability of being obese at time t, G is 1 for females and 0 for males and t equals 1, 2 or 3. Thus the odds of being obese are, for the three evaluation times are,

```
> for (i in 1:3) {
+     print(round(exp(-0.774 + 0.312 * i), 3))
+ }
```

```
[1] 0.63
[1] 0.861
[1] 1.176
```

According to this the odds of being obese increase through time for females.

To fit the same data with a Markov–chain model we need to create a new data frame.

```
> tb9.13b <- data.frame(expand.grid(gend = c("male", "female"),
+     prev1 = c("N", "O"), time = c(79, 81)))
> tb9.13b$O <- c(sum(tb9.13[tb9.13$resp.79 == 1 & tb9.13$gend ==
+     0 & tb9.13$resp.77 == 0, "count"]), sum(tb9.13[tb9.13$resp.79 ==
+     1 & tb9.13$gend == 1 & tb9.13$resp.77 == 0, "count"]), sum(tb9.13[tb9.13$resp.79 ==
+     1 & tb9.13$gend == 0 & tb9.13$resp.77 == 1, "count"]), sum(tb9.13[tb9.13$resp.79 ==
+     1 & tb9.13$gend == 1 & tb9.13$resp.77 == 1, "count"]), sum(tb9.13[tb9.13$resp.81 ==
+     1 & tb9.13$gend == 0 & tb9.13$resp.79 == 0, "count"]), sum(tb9.13[tb9.13$resp.81 ==
+     1 & tb9.13$gend == 1 & tb9.13$resp.79 == 0, "count"]), sum(tb9.13[tb9.13$resp.81 ==
+     1 & tb9.13$gend == 0 & tb9.13$resp.79 == 1, "count"]), sum(tb9.13[tb9.13$resp.81 ==
+     1 & tb9.13$gend == 1 & tb9.13$resp.79 == 1, "count"]))
> tb9.13b$N <- c(sum(tb9.13[tb9.13$resp.79 == 0 & tb9.13$gend ==
```

```
+     0 & tb9.13$resp.77 == 0, "count"]), sum(tb9.13[tb9.13$resp.79 ==
+     0 & tb9.13$gend == 1 & tb9.13$resp.77 == 0, "count"]), sum(tb9.13[tb9.13$resp.79 ==
+     0 & tb9.13$gend == 0 & tb9.13$resp.77 == 1, "count"]), sum(tb9.13[tb9.13$resp.79 ==
+     0 & tb9.13$gend == 1 & tb9.13$resp.77 == 1, "count"]), sum(tb9.13[tb9.13$resp.81 ==
+     0 & tb9.13$gend == 0 & tb9.13$resp.79 == 0, "count"]), sum(tb9.13[tb9.13$resp.81 ==
+     0 & tb9.13$gend == 1 & tb9.13$resp.79 == 0, "count"]), sum(tb9.13[tb9.13$resp.81 ==
+     0 & tb9.13$gend == 0 & tb9.13$resp.79 == 1, "count"]), sum(tb9.13[tb9.13$resp.81 ==
+     0 & tb9.13$gend == 1 & tb9.13$resp.79 == 1, "count"]))
> tb9.13b <- within(tb9.13b, {
+     total <- O + N
+     prop.N <- N/total
+     prop.O <- O/total
+ })
> tb9.13b$time2 <- ifelse(tb9.13b$time == 79, 1, 2)
```

Note, again, that a second `time` variable was introduced.
Then we fit the model,

$$logit[P(Y_t = 1)] = \alpha + \beta_1 Prev + \beta_2 t + \beta_3 G + \beta_4 t \times G$$

Where $P(Y_t = 1)$ is the probability of being obese, $Prev$ equals 1 for a previous "obese" response, t equals 1 or 2 depending on the observation (Note that in this case 2 refers to the last observation) and G equals 1 for females.

```
> summary(fit2.glm13 <- glm(prop.O ~ prev1 + time2 * gend, weights = total,
+     family = binomial, data = tb9.13b))$coefficients
```

```
                  Estimate Std. Error    z value     Pr(>|z|)
(Intercept)      -2.1782884  0.5127266 -4.2484407 2.152636e-05
prev1O            2.8936222  0.2310478 12.5239131 5.524291e-36
time2            -0.2563498  0.3231531 -0.7932767 4.276166e-01
gendfemale        0.5152234  0.7060574  0.7297188 4.655621e-01
time2:gendfemale -0.1645889  0.4532007 -0.3631700 7.164779e-01
```

Note that except for the previous response all the other explanatory variables have non-significant p-values; this may be due to the fact that when a previous response variable is included the magnitude of the other explanatory variables decreases (see Agresti 2007, p. 289).
Because of this some explanatory variables were removed, based on the AIC value, this with the `stepAIC` function from the `MASS` package.

```
> summary(stepAIC(fit2.glm13, direction = "backward", trace = FALSE,
+     list(lower = ~1, upper = formula(fit2.glm13)), scale = 1))$coefficients
```

```
             Estimate Std. Error   z value     Pr(>|z|)
(Intercept) -1.9170603  0.3571635 -5.367459 7.985371e-08
prev1O       2.8582214  0.2275274 12.562095 3.411776e-36
time2       -0.3310543  0.2264424 -1.461980 1.437467e-01
```

This procedure selects a model with only the `previous response` and the `time effect`. However note that the p-value for the `time` effect is still above the cut point of 0.05. Despite this, conditional on the time the odds of being obese are $exp(2.86) = 17.462$ times that when the previous response was normal. Conditional on the previous response, the odds of being obese at the two last evaluations are,

```
> for (i in 1:2) {
+     print(round(exp(-0.331 * i), 3))
+ }
```

```
[1] 0.718
[1] 0.516
```

that of having a normal weight.

Problem 16

First we construct the data frame which is basically the ungrouped data used in chapter 6, however in this case the LDL levels are re-codified and converted to a numeric vector.

```
> tb6.19 <- read.table('supp_data/tb6-19', header = TRUE)
> tb6.19long2 <- tb6.19[rep(1:nrow(tb6.19), tb6.19$values), ]
> tb6.19long2 <- within(tb6.19long2, {
+     rm(values)
+     case <- 1:370
+ })
> tb6.19long2 <- tb6.19long2[order(tb6.19long2$case), ]
> tb6.19long2$ldl2 <- ordered(tb6.19long2$ldl2, levels = 1:4)
```

Then we fit the model

$$logit[P(Y_2 \leq j)] = \alpha + beta_1 T + \beta_2 LDL1_j$$

Where $P(Y_2 \leq j)$ is the probability that the second measurement of LDL would be lower or equal to a specified value, T is 1 for the new treatment and LDL_j corresponds to the first measure of LDL.

```
> summary(fit.ordgee16 <- ordgee(ldl2 ~ ldl1 + Treatment, id = case,
+     data = tb6.19long2, corstr = "independence", control = geese.control(maxit = 100),
+     rev = TRUE))
```

```
Call:
ordgee(formula = ldl2 ~ ldl1 + Treatment, id = case, data = tb6.19long2,
    corstr = "independence", control = geese.control(maxit = 100),
    rev = TRUE)

Mean Model:
 Mean Link:                 logit
 Variance to Mean Relation: binomial

 Coefficients:
                     estimate    san.se      wald            p
Inter:1             1.9693991 1.2364680  2.536891 1.112136e-01
Inter:2             4.3130524 1.1128757 15.020215 1.063657e-04
Inter:3             6.8950321 1.0943353 39.698289 2.963850e-10
ldl1               -1.6470426 0.4217658 15.249897 9.418191e-05
TreatmentTreatment  0.6730752 0.4317061  2.430809 1.189715e-01
```

```
Scale is fixed.

Correlation Model:
 Correlation Structure:     independence

Returned Error Value:    0
Number of clusters:   370   Maximum cluster size: 1
```

According to these results the most important explanatory variable is the previous measure of LDL and there seem to be no differences between the treatments.

Problem 17

As it's mentioned in page 268 the correct statement would be that given Y_{t-1}, in a first order Markov–chain model, Y_t is *conditionally independent* (not marginal independent) of Y_{t-2} (see Agresti 2007, p. 53 for a definition of conditional and marginal independence) .

Problem 18

True, see page 287.

References

Agresti, Alan (2007). *An Introduction to Categorical Data Analysis*. 2nd ed. Hoboken NJ: Wiley-Interscience. ISBN: 978-0-471-22618-5.

Thompson, Laura A. (2007). *S-plus (and R) Manual to Accompany Agresti's "Categorical Data Analysis" (2002)*.