STSB6806 Test 2 of 2021

Mathematical Statistics and Actuarial Science; University of the Free State

2021/06/10

Time: 180 minutes; Marks: 50

MEMORANDUM

Instructions

- Answer all questions in a single R Markdown document. Please knit to PDF or Word at the end and submit both the PDF/Word document and the .Rmd file for assessment, in that order.
- Label questions clearly, as it is done on this question paper.
- All results accurate to 4 decimal places.
- Show all derivations, formulas, code, sources and reasoning.
- Intervals should cover 95% probability unless stated otherwise.
- No communication software, no devices, and no communication capable websites may be accessed prior to submission. You may not (nor even appear to) attempt to communicate or pass information to another student.

Question 1

Your task in this question is to fit a simplified analysis on data exported from the paper:

Worldwide trends in blood pressure from 1975 to 2015: a pooled analysis of 1479 population-based measurement studies with 19·1 million participants by Zhou, Bin *et al.* published in The Lancet , Volume 389 , Issue 10064 , 37 – 55.

The data is provided as-is on https://ufs.blackboard.com. It is a summary prepared by the authors on the world-wide averages of blood pressure measurements done from 1975 to 2015. Your data is summarised into annual averages, separately for men and women.

We will focus on the prevalence of high blood pressure measurements.

1.1) Read in the data and give a summary of the 5 columns. Then convert the prevalence values (with intervals) to a free scale using a logistic transform and give a new summary. **[5]**

```
blooddata <- read.csv('BloodPressureData.csv')
summary(blooddata)
```

```
|    Gender           Year      PrevalenceOfRaisedBloodPressure
|  Length:82        Min.   :1975   Min.   :0.1888
|  Class :character   1st Qu.:1985   1st Qu.:0.2273
|  Mode  :character   Median :1995   Median :0.2483
|                     Mean   :1995   Mean   :0.2518
|             3rd Qu.:2005   3rd Qu.:0.2736
|             Max.   :2015   Max.   :0.3690
|     Lower95          Upper95
|  Min.   :0.1429   Min.   :0.2359
|  1st Qu.:0.1838   1st Qu.:0.2716
|  Median :0.2006   Median :0.3091
|  Mean   :0.2012   Mean   :0.3110
|  3rd Qu.:0.2200   3rd Qu.:0.3436
|  Max.   :0.2749   Max.   :0.5049
```

```
Gender <- factor(blooddata[,1])
Year <- blooddata[,2]
Orig.Prev <- blooddata[,3]
Orig.Lower <- blooddata[,4]
Orig.Upper <- blooddata[,5]
Free.Prev <- qlogis(Orig.Prev)
Free.Lower <- qlogis(Orig.Lower)
Free.Upper <- qlogis(Orig.Upper)
summary(d <- data.frame(Gender,Year,Free.Prev,Free.Lower,Free.Upper))
```

```
|    Gender        Year         Free.Prev          Free.Lower
|  Men  :41   Min.   :1975   Min.   :-1.4578   Min.   :-1.7916
|  Women:41   1st Qu.:1985   1st Qu.:-1.2236   1st Qu.:-1.4905
|             Median :1995   Median :-1.1075   Median :-1.3822
|             Mean   :1995   Mean   :-1.0969   Mean   :-1.3882
|             3rd Qu.:2005   3rd Qu.:-0.9764   3rd Qu.:-1.2654
|             Max.   :2015   Max.   :-0.5365   Max.   :-0.9699
|    Free.Upper
|  Min.   :-1.17555
|  1st Qu.:-0.98676
|  Median :-0.80410
|  Mean   :-0.80547
|  3rd Qu.:-0.64709
|  Max.   : 0.01964
```

Load data [1], Summary [1], Transform [2], Summary [1]

Work with the free scale from now on, except where otherwise indicated.

1.2)
Assuming an underlying Gaussian distribution for the intervals and that they are symmetric (2.5% to 97.5%), estimate the standard deviation for each row in your table. Show that the median standard deviation is about 0.13. **[3]**

```
sd.ests <- (Free.Upper-Free.Lower)/2/qnorm(0.975)
median(sd.ests)
```

```
| [1] 0.1300055
```

Transform correctly [3]

1.3)
'Standardise' the Year column by dividing by 2000. Keep the transformation in mind when drawing plots and conclusions. You may also convert Gender to 1 versus 2 for easy modelling. Why is standardising useful for Bayes models? **[3]**

```
SYear <- Year/2000
GenderNum <- as.numeric(Gender)
```

Transform year [1], transform gender [1], A sensible reason [1], e.g. it makes prior and initial value assignment easier or automatic choices more valid.

1.4)
Fit an ordinary Bayesian regression line for the prevalence on the free scale, ignoring gender entirely. The model is as follows:

$$Y_i \sim N(\mu_i, \sigma_i^2)\,, \quad \mu_i = \beta_0 + \beta_1 Year_i\,, \quad \beta_0, \beta_1 \sim N(0, 100^2)$$

Where σ_i is obtained from the previous question.

Determine the significance of each of the β parameters at $\alpha = 0.02$. **[8]**

```
library(rstan)
mycores <- 3
options(mc.cores = mycores)
```

```
// This Stan block defines a linear model with known variation, by Sean van der Merwe, UFS
data {
  int<lower=1> n;                 // number of observations
  real y[n];      // observations
  real s[n];      // standard deviations of observations
  real x[n];      // explanatory variable
}
parameters {
  real beta0;             // intercept
  real beta1;             // slope
}
transformed parameters {
  real mu[n];     // expected values
  for (i in 1:n) {
    mu[i] = beta0 + beta1*x[i];
  }
}
model {
  y ~ normal(mu, s);      // fit the data pattern
  beta0 ~ normal(0, 100);
  beta1 ~ normal(0, 100);
}
generated quantities {
  vector[n] log_lik;
  for (i in 1:n) {
    log_lik[i] = normal_lpdf(y[i] | mu[i], s[i]);
  }
}
```

```
saveRDS(LM1, file = 'LM1.Rds')
```

```
stan_data1 <- list(n=length(Free.Prev), y=Free.Prev, x=SYear, s=sd.ests)
ModelFit1 <- sampling(LM1, stan_data1, pars=c('beta0', 'beta1', 'log_lik'), iter =
10000, chains = mycores, control=list(max_treedepth=15))
```

```
saveRDS(ModelFit1, file = 'LM1sim.Rds')
```

```
draws1 <- extract(ModelFit1)
kable(round(summary(ModelFit1)$summary[1:2,],3))
```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat

beta0	13.392	0.07	3.005	7.498	11.331	13.429	15.463	19.119	1843.346	1.001
beta1	-14.528	0.07	3.006	-20.265	-16.600	-14.568	-12.475	-8.627	1843.164	1.001

```
pvalfunc <- function(sims,target=0) { 2*min(mean(sims<target),mean(sims>target)) }
cat('The significance can be seen from the simulation summary, and the p-value
equivalents are', pvalfunc(draws1$beta0), 'and', pvalfunc(draws1$beta1))

| The significance can be seen from the simulation summary, and the p-value equivalents are 0 and 0
```

Model code correct [3], including priors [1], data correctly loaded [2], All betas correctly determined as significant.[2]

1.5) Fit an ordinary Bayesian regression line for the prevalence on the free scale, with each gender having its own line. The model is as follows:

$$Y_{ig} \sim N\left(\mu_{ig}, \sigma_{ig}^2\right), \quad \mu_{ig} = \beta_{0g} + \beta_{1g} Year_i\,, \quad \beta_{0g}, \beta_{1g} \sim N(0, 100^2)$$

Determine the significance of each of the β parameters at $\alpha = 0.01$. Also give a p-value for the hypothesis $\beta_{1f} = \beta_{1m}$. **[7]**

```
// This Stan block defines a linear model with known variation, by Sean van der Merwe, UFS
data {
 int<lower=1> n;                  // number of observations
 real y[n];      // observations
 real s[n];      // standard deviations of observations
 real x[n];      // explanatory variable
 int<lower=1> ng;                  // number of groups
 int<lower=1, upper=ng> g[n];    // group membership
}
parameters {
 real beta0[ng];              // intercept
 real beta1[ng];              // slope
}
transformed parameters {
 real mu[n];      // expected values
 for (i in 1:n) {
  mu[i] = beta0[g[i]] + beta1[g[i]]*x[i];
 }
}
model {
 y ~ normal(mu, s);      // fit the data pattern
 beta0 ~ normal(0, 100);
 beta1 ~ normal(0, 100);
}
generated quantities {
 vector[n] log_lik;
 for (i in 1:n) {
  log_lik[i] = normal_lpdf(y[i] | mu[i], s[i]);
 }
}

saveRDS(LM2, file = 'LM2.Rds')

stan_data2 <- list(n=length(Free.Prev), y=Free.Prev, x=SYear, s=sd.ests,
ng=max(GenderNum), g=GenderNum)
ModelFit2 <- sampling(LM2, stan_data2, pars=c('beta0', 'beta1', 'log_lik'), iter =
10000, chains = mycores, control=list(max_treedepth=15))
```

```
saveRDS(ModelFit2, file = 'LM2sim.Rds')

draws2 <- extract(ModelFit2)
kable(round(summary(ModelFit2)$summary[1:4,],3))
```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
beta0[1]	9.322	0.069	4.527	0.703	6.250	9.245	12.448	18.012	4360.611	1
beta0[2]	17.148	0.065	4.251	8.805	14.261	17.186	20.024	25.424	4215.314	1
beta1[1]	-10.365	0.069	4.529	-19.061	-13.490	-10.286	-7.286	-1.729	4360.964	1
beta1[2]	-18.367	0.066	4.253	-26.636	-21.244	-18.408	-15.478	-10.032	4215.262	1

```
kable(data.frame(Parameter=c('beta0 Men', 'beta0 Women', 'beta1 Men', 'beta1
Women'), pvalue=c(apply(draws2$beta0, 2, pvalfunc), apply(draws2$beta1, 2,
pvalfunc))))
```

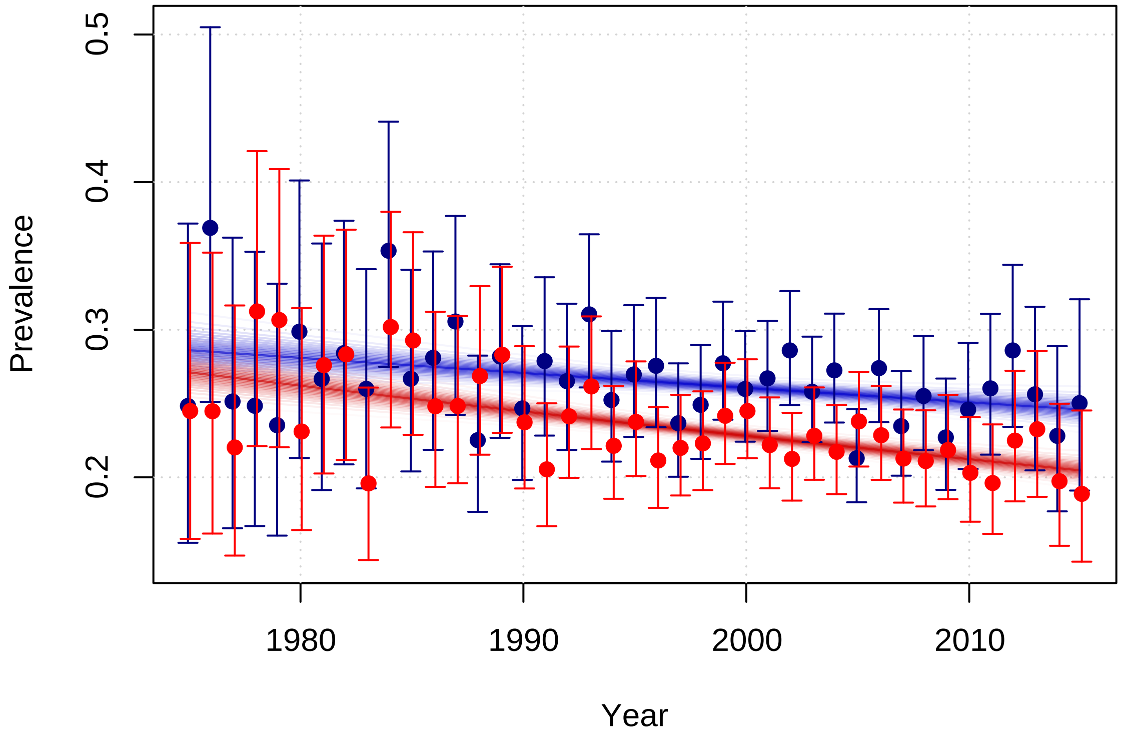
Parameter	pvalue
beta0 Men	0.034
beta0 Women	0.000
beta1 Men	0.020
beta1 Women	0.000

```
cat('Test of difference in slopes has p-value:', pvalfunc(draws2$beta1[,2]-
draws2$beta1[,1]))

| Test of difference in slopes has p-value: 0.1973333
```

Model code correctly adapted [3], simulation parameters good [1], All coefficients correctly determined as significant or not, versus 0.01 [2], but difference in slopes not significant [1]

1.6) Draw a plot (on the original scale ideally) showing your model fit lines, with uncertainty. Overlay the observed values, also with uncertainty ideally. Comment on how well the models fit the observations in your opinion. **[8]**

```
par(mar=c(4.5, 4.5, 0.5, 0.5))
minyear <- min(Year)
maxyear <- max(Year)
yearseq <- minyear:maxyear
plot(c(minyear, maxyear), c(min(Orig.Lower), max(Orig.Upper)), type='n', main='',
xlab='Year', ylab='Prevalence')
grid()
qseq <- seq(2,50,4)/100; qseq <- c(matrix(c(qseq,1-qseq), 2, length(qseq), byrow =
TRUE))
preds1 <- sapply(yearseq, function(y) {
  quantile(plogis(draws2$beta0[,1] + draws2$beta1[,1]*y/2000), qseq)
})
preds2 <- sapply(yearseq, function(y) {
  quantile(plogis(draws2$beta0[,2] + draws2$beta1[,2]*y/2000), qseq)
})
for (i in seq_along(qseq)) {
  lines(yearseq, preds1[i,], col=rgb(0,0,0.8,(i/2/length(qseq))))
  lines(yearseq, preds2[i,], col=rgb(0.8,0,0,(i/2/length(qseq))))
}
for (j in yearseq) {
  arrows(j-0.05, Orig.Lower[(Year==j) & (GenderNum==1)], j-0.05,
Orig.Upper[(Year==j) & (GenderNum==1)], col='navy', angle=90, length=0.05, code=3)
  points(j-0.05, Orig.Prev[(Year==j) & (GenderNum==1)], col='navy', pch=19)
  arrows(j+0.05, Orig.Lower[(Year==j) & (GenderNum==2)], j+0.05,
Orig.Upper[(Year==j) & (GenderNum==2)], col='red', angle=90, length=0.05, code=3)
  points(j+0.05, Orig.Prev[(Year==j) & (GenderNum==2)], col='red', pch=19)
}
```



Plot of model lines [2], with uncertainty [2], data values [2], with uncertainty [2]

1.7) Adapt the model to have parallel slopes across genders, replace β_{1g} with β_1. Determine the significance of β_1 visually. **[4]**

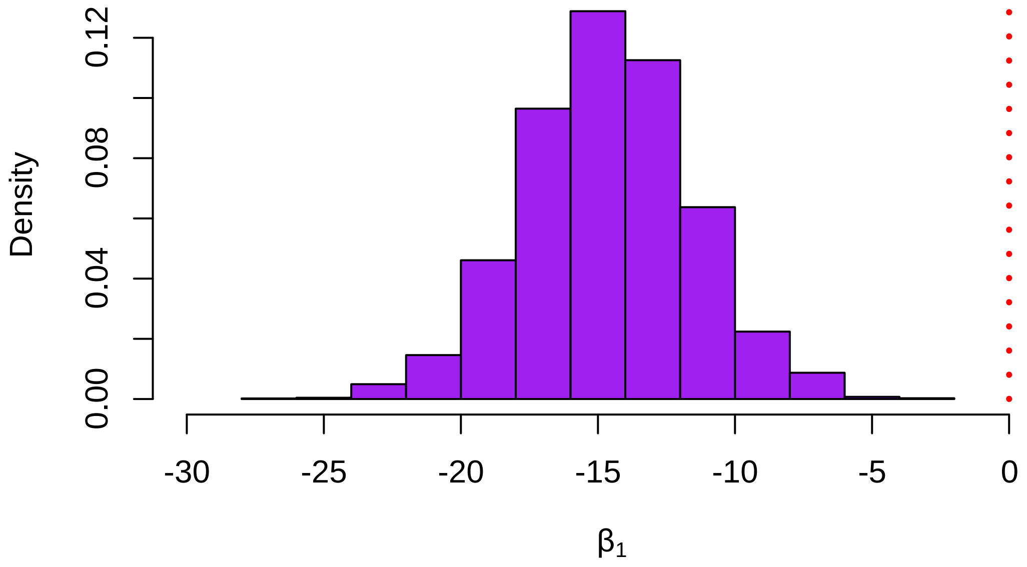
```
// This Stan block defines a linear model with known variation, by Sean van der Merwe, UFS
data {
  int<lower=1> n;                    // number of observations
  real y[n];      // observations
  real s[n];      // standard deviations of observations
  real x[n];      // explanatory variable
  int<lower=1> ng;                    // number of groups
  int<lower=1, upper=ng> g[n];    // group membership
}
parameters {
  real beta0[ng];              // intercept
  real beta1;              // slope
}
transformed parameters {
  real mu[n];    // expected values
  for (i in 1:n) {
    mu[i] = beta0[g[i]] + beta1*x[i];
```

```
 }
}
model {
 y ~ normal(mu, s);      // fit the data pattern
 beta0 ~ normal(0, 100);
 beta1 ~ normal(0, 100);
}
generated quantities {
 vector[n] log_lik;
 for (i in 1:n) {
  log_lik[i] = normal_lpdf(y[i] | mu[i], s[i]);
 }
}
```

```
saveRDS(LM3, file = 'LM3.Rds')

ModelFit3 <- sampling(LM3, stan_data2, pars=c('beta0', 'beta1', 'log_lik'), iter =
10000, chains = mycores, control=list(max_treedepth=15))

saveRDS(ModelFit3, file = 'LM3sim.Rds')

draws3 <- extract(ModelFit3)
kable(round(summary(ModelFit3)$summary[1:3,],3))
```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
beta0[1]	13.581	0.086	3.076	7.395	11.523	13.623	15.633	19.538	1275.992	1.005
beta0[2]	13.407	0.086	3.076	7.239	11.350	13.447	15.467	19.376	1275.721	1.005
beta1	-14.625	0.086	3.077	-20.592	-16.682	-14.663	-12.568	-8.444	1275.764	1.005

```
hist(draws3$beta1, main='', xlab = expression(beta[1]), col='purple', freq=FALSE,
xlim=c(-30,1))
lines(c(0,0), c(0,1), lwd=3, lty=3, col='red')
```



Changing the model correctly [2], drawing a histogram, density, or box plot of beta1 [1], commenting that it is one side of 0 [1].

1.8) Also give a p-value for the hypothesis $\beta_{0f} = \beta_{0m}$, and explain why it is an apparent contradiction to the intervals of β_{0f} and β_{0m}. **[3]**

```r
cat('Test of difference in intercepts has p-value:', pvalfunc(draws3$beta0[,2]-
draws3$beta0[,1]))
```

```
| Test of difference in intercepts has p-value: 0
```

```r
shortestinterval <- function(postsims,alpha=0.05) { # Coded by Sean van der Merwe,
UFS
sorted.postsims <- sort(postsims)
nsims <- length(postsims)
gap <- round(nsims*(1-alpha))
widths <- diff(sorted.postsims,gap)
interval <- sorted.postsims[c(which.min(widths),(which.min(widths) + gap))]
return(interval) }

int_table <- apply(draws3$beta0, 2, shortestinterval)
rownames(int_table) <- c('Lower','Upper'); colnames(int_table) <- levels(Gender)
kable(round(int_table,1))
```

	Men	Women
Lower	7.5	7.3
Upper	19.6	19.4

Correct p-value [1], noting that the intervals are similar and have large overlap [1], explaining that the reason for the contradiction is that the intercepts have massive correlation [1]. Basically, the model knows that the intercepts should be different but does not know what they should be.

1.9) Compare all the models you fitted using either LOOIC, DIC, or Bayes Factors. Once you have determined what you consider to be the best model of those you fitted, explain what this result implies about the data and data generating process. **[9]**

```
library(loo)
log_lik_1 <- extract_log_lik(ModelFit1, merge_chains = FALSE)
log_lik_2 <- extract_log_lik(ModelFit2, merge_chains = FALSE)
log_lik_3 <- extract_log_lik(ModelFit3, merge_chains = FALSE)
r_eff_1 <- relative_eff(exp(log_lik_1), cores = mycores)
r_eff_2 <- relative_eff(exp(log_lik_2), cores = mycores)
r_eff_3 <- relative_eff(exp(log_lik_3), cores = mycores)
loo_1 <- loo(log_lik_1, r_eff = r_eff_1, cores = mycores)
loo_2 <- loo(log_lik_2, r_eff = r_eff_2, cores = mycores)
loo_3 <- loo(log_lik_3, r_eff = r_eff_3, cores = mycores)
comp <- loo_compare(loo_1, loo_2, loo_3)
print(comp, simplify=FALSE)
```

```
|        elpd_diff se_diff elpd_loo se_elpd_loo p_loo  se_p_loo looic  se_looic
| model3   0.0      0.0    51.4      5.5        2.4    0.4   -102.9   11.0
| model2  -0.1      1.2    51.4      5.7        3.1    0.5   -102.8   11.4
| model1 -18.7      5.9    32.7      6.8        2.5    0.3    -65.4   13.5
```

Code for calculating a statistics from each model [2x3=6]. Reasonable comparison and conclusion [3], e.g. LOOIC is the smallest for the model with two intercepts and two slopes, suggesting that as the most parsimonious model.

1.10) [Bonus] Adapt the model to use an MA(1) or AR(1) model over time instead of a linear model. One possibility is $\mu_{(i)g} = \beta_{0g}$ if $i = 1$ and $\mu_{(i)g} = \theta_g(y_{(i-1)g} - \mu_{(i-1)g})$ if $i > 1$. Use standard normal priors for θ_g. Calculate $P[\theta_1 > \theta_2 > 0]$. [Do not attempt until all other questions are answered, maximum +6 marks]

```
// This Stan block defines a linear model with known variation, by Sean van der Merwe, UFS
data {
 int<lower=1> n;                    // number of observations
 real y[n,2];      // observations
 real s[n,2];      // standard deviations of observations
}
parameters {
 real beta0[2];              // intercepts
 real theta[2];             // correlations
}
transformed parameters {
 real mu[n,2];      // expected values
 mu[1,1] = beta0[1];
 mu[1,2] = beta0[2];
 for (i in 2:n) {
  mu[i,1] = theta[1]*(y[(i-1),1]-mu[(i-1),1]);
  mu[i,2] = theta[2]*(y[(i-1),2]-mu[(i-1),2]);
 }
}
model {
 for (j in 1:2) {
  for (i in 1:n) {
   y[i,j] ~ normal(mu[i,j], s[i,j]);      // fit the data pattern
  }
  beta0[j] ~ normal(0, 100);
  theta[j] ~ normal(0, 1);
 }
}
generated quantities {
 vector[n*2] log_lik;
 for (i in 1:n) {
  log_lik[i] = normal_lpdf(y[i,1] | mu[i,1], s[i,1]);
  log_lik[i+n] = normal_lpdf(y[i,2] | mu[i,2], s[i,2]);
 }
}

saveRDS(LM4lf, file = 'LM4.Rds')

nyears <- length(yearseq)
stan_data4 <- list(n=nyears, y=matrix(Free.Prev,nyears), x=SYear,
s=matrix(sd.ests,nyears))
ModelFit4 <- sampling(LM4, stan_data4, pars=c('beta0', 'theta', 'log_lik'), iter =
10000, chains = mycores, control=list(max_treedepth=15))

saveRDS(ModelFit4, file = 'LM4sim.Rds')

draws4 <- extract(ModelFit4)
kable(round(summary(ModelFit4)$summary[1:4,],3))
```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
beta0[1]	-0.855	0.291	0.357	-1.150	-1.111	-1.091	-0.359	-0.338	1.502	65.585
beta0[2]	-0.779	0.001	0.001	-0.781	-0.780	-0.779	-0.778	-0.778	1.640	4.728
theta[1]	1.242	0.261	0.320	0.781	0.798	1.428	1.515	1.517	1.510	25.325
theta[2]	1.563	0.020	0.026	1.524	1.540	1.566	1.586	1.603	1.630	4.529

```
mean((draws4$theta[,1]>draws4$theta[,2]) & (draws4$theta[,2] > 0))

| [1] 0
```

Transform model correctly [3], with correct priors [1], run correctly [1], calculate joint probability [1].

Points total

The points on the test add up to **50**