STAT 578 - Advanced Bayesian Modeling - Fall 2019 Assignment 4

Xiaoming Ji

Solution for Problem (a)

(i)

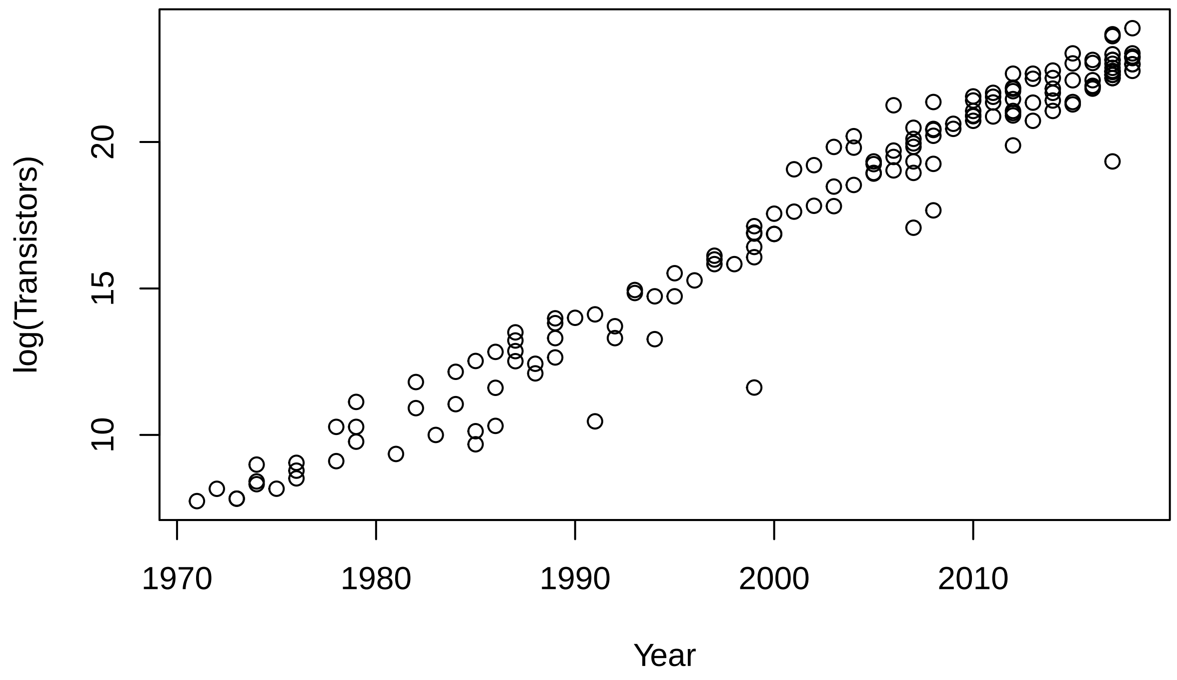
According to the formula for Moore's law and take log on T, we get:

$log(T) \approx log(C * 2^{A/2}) \rightarrow log(T) \approx log(C) + 0.3465736 * A$

This is the form of linear regression that make logT as response variable and A as explanatory variable. `log(C)` is the intercept and coefficient of A is 0.3465736. Thus we can say logT is roughly follow a simple linear regression on A.

(ii)

```r
moores_df = read.csv("mooreslawdata.csv", header=TRUE)
with(moores_df, plot(log(Transistors) ~ Year))
```



Solution for Problem (b)

(i)

We first use classical linear model to estimate the parameters.

```
df_lm <- list(y = log(moores_df$Transistors),
              x_centered = moores_df$Year - mean(moores_df$Year))

model_lm = lm(y ~ x_centered, data=df_lm)
summary(model_lm)
```

```
##
## Call:
## lm(formula = y ~ x_centered, data = df_lm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0598 -0.4443  0.0891  0.6052  2.1820
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.726825   0.074044  239.41   <2e-16 ***
## x_centered   0.342122   0.005436   62.93   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9395 on 159 degrees of freedom
## Multiple R-squared:  0.9614, Adjusted R-squared:  0.9612
## F-statistic:  3960 on 1 and 159 DF,  p-value: < 2.2e-16
```

- β estimates are about 0.3 to 18, we choose to set initial β values at ± 200
- Regression error variance σ^2 estimate is about $(0.94^2) \approx 0.9$, we choose to set initial σ^2 values of 0.01 and 100

```
model {
  x_bar <- mean(x)
  for (i in 1:length(y)) {
    y[i] ~ dnorm(beta1 + beta2* (x[i] - x_bar), sigmasqinv)
  }

  beta1 ~ dnorm(0, 1e-06)
  beta2 ~ dnorm(0, 1e-06)
  sigmasqinv ~ dgamma(0.001, 0.001)
  sigmasq <- 1/sigmasqinv
}
```

```
library(rjags)

df_jags <- list(y = log(moores_df$Transistors),
                x = moores_df$Year)

initial_vals <- list(list(beta1 = 200, beta2 = 200, sigmasqinv = 100),
                     list(beta1 = 200, beta2 = -200, sigmasqinv = 100),
                     list(beta1 = -200, beta2 = 200, sigmasqinv = 0.01),
                     list(beta1 = -200, beta2 = -200, sigmasqinv = 0.01))
```

```r
moores_model <- jags.model("moores.bug", df_jags, initial_vals, n.chains = 4)
update(moores_model, 1000)
coef_sample <- coda.samples(moores_model, c("beta1","beta2","sigmasq"), n.iter=2000)
```

```r
gelman.diag(coef_sample, autoburnin=FALSE)
```

```
## Potential scale reduction factors:
##
##         Point est. Upper C.I.
## beta1            1          1
## beta2            1          1
## sigmasq          1          1
##
## Multivariate psrf
##
## 1
```

Gelman-Rubin statistic for $\beta1$, $\beta2$ and σ^2 are all 1, thus we can declare convergence of them.

(ii)

```r
summary(coef_sample)
```

```
##
## Iterations = 1001:3000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##            Mean       SD  Naive SE Time-series SE
## beta1   17.7271 0.074781 0.0008361      8.362e-04
## beta2    0.3421 0.005456 0.0000610      6.014e-05
## sigmasq  0.8924 0.099704 0.0011147      1.111e-03
##
## 2. Quantiles for each variable:
##
##            2.5%     25%     50%     75%   97.5%
## beta1   17.5780 17.6780 17.7269 17.7779 17.8729
## beta2    0.3313  0.3385  0.3422  0.3457  0.3527
## sigmasq  0.7193  0.8231  0.8833  0.9550  1.1128
```

(iii)

```r
beta2_sample = as.matrix(coef_sample)[,"beta2"]
```

Mean of slope,

```r
mean(beta2_sample)
```

```
## [1] 0.3420985
```

95% posterior credible interval of slope,

```r
quantile(beta2_sample, c(0.025, 0.975))
```

```
##     2.5%    97.5% 
## 0.331320 0.352727
```

The interval contains value (0.3465736) determined in part (a).

(iv)

```r
beta1_sample = as.matrix(coef_sample)[,"beta1"]
```

Mean of intercept,

```r
mean(beta1_sample)
```

```
## [1] 17.72707
```

95% posterior credible interval of intercept,

```r
quantile(beta1_sample, c(0.025, 0.975))
```

```
##     2.5%    97.5% 
## 17.57802 17.87290
```

Solution for Problem (c)

(i)

```
model {
  x_bar <- mean(x)
  for (i in 1:length(y)) {
    y[i] ~ dnorm(beta1 + beta2* (x[i] - x_bar), sigmasqinv)
  }
  beta1 ~ dnorm(0, 1e-06)
  beta2 ~ dnorm(0, 1e-06)
  sigmasqinv ~ dgamma(0.001, 0.001)
  y_predict ~ dnorm(beta1 + beta2*(year_predict - x_bar), sigmasqinv)
  year_start <- x_bar - (beta1 / beta2)
}
```

```r
predict_model <- jags.model("moores_predict.bug",
                            c(as.list(df_jags), year_predict = 2020),
                            initial_vals, n.chains = 4)
update(predict_model, 1000)
predict_sample <- coda.samples(predict_model, c("y_predict", "year_start"), n.iter=2000)
```

```r
gelman.diag(predict_sample, autoburnin=FALSE)
```

```
## Potential scale reduction factors:
## 
##            Point est. Upper C.I.
## y_predict           1          1
## year_start          1          1
## 
```

```
## Multivariate psrf
##
## 1
```

Gelman-Rubin statistic for the prediction is 1, thus we can declare convergence of them.

(ii)

```
summary(predict_sample)
```

```
##
## Iterations = 1001:3000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                Mean     SD Naive SE Time-series SE
## y_predict    23.84 0.9490  0.01061       0.010571
## year_start 1950.25 0.8586  0.00960       0.009488
##
## 2. Quantiles for each variable:
##
##            2.5%    25%     50%     75%   97.5%
## y_predict    22   23.2   23.84   24.49   25.72
## year_start 1949 1949.7 1950.26 1950.85 1951.89
```

(iii)

95% posterior predictive interval for the transistor count, in billions,

```
exp(quantile(as.matrix(predict_sample)[,"y_predict"], c(0.025, 0.975))) / (10^9)
```

```
##       2.5%      97.5%
##   3.588029 147.620426
```

(iv)

To explain, we can assume the year when transistor is in invented is the year for transistor count equal to 1. This means given our model,

$log(T) = \beta 1 + \beta 2 * (A_i - \bar{A})$

We can set T = 1 to derive A_i as the year when transistor is invented. Therefore, we have,

$A_i = \bar{A} - \beta 1/\beta 2$

Using the classical linear model, we can estimate this year as,

```
round(mean(moores_df$Year) - coef(model_lm)[[1]]/coef(model_lm)[[2]])
```

```
## [1] 1950
```

The 95% posterior interval for this quantity.

```r
round(quantile(as.matrix(predict_sample)[,"year_start"], c(0.025, 0.975)))
```

```
##  2.5% 97.5%
##  1949  1952
```

Note: According to Wikipedia, the first working device to be built was a point-contact transistor invented in *1947* by American physicists John Bardeen, Walter Brattain, and William Shockley at Bell Labs. Our estimates is very close.

Solution for Problem (d)

(i)

```r
X <- model.matrix(model_lm)
Nsim <- length(beta1_sample)
error_sim <- matrix(NA, Nsim, nrow(X))
post_beta <- as.matrix(coef_sample)[, c("beta1", "beta2")]
post_simgasq <- as.matrix(coef_sample)[, c("sigmasq")]

for(s in 1:Nsim) {
  error_sim[s,] <- df_jags$y - X %*% cbind(post_beta[s,])
}
```

(ii)

```r
error_rep <- matrix(NA, Nsim, nrow(X))

for(s in 1:Nsim) {
  error_rep[s,] <- rnorm(nrow(X), 0, sqrt(post_simgasq[s]))
}
```

(iii)

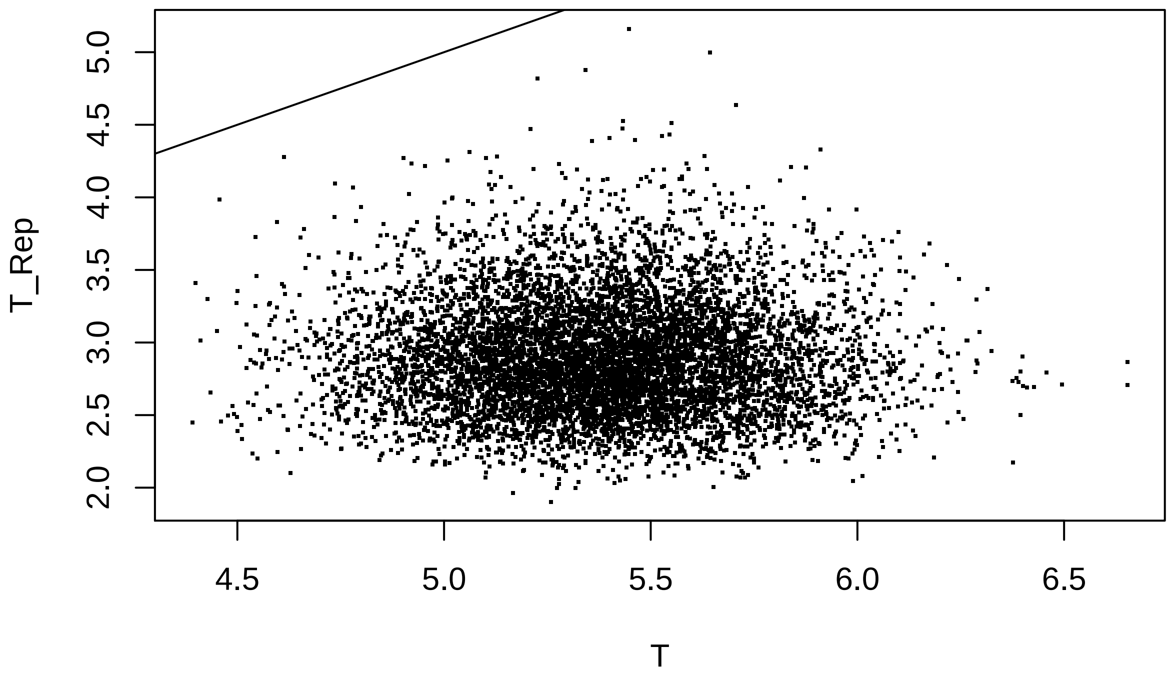
```r
error_sim_std <- matrix(NA, Nsim, nrow(X))
error_rep_std <- matrix(NA, Nsim, nrow(X))

for(s in 1:Nsim) {
  error_sim_std[s,] = error_sim[s,] / sqrt(post_simgasq[s])
  error_rep_std[s,] = error_rep[s,] / sqrt(post_simgasq[s])
}

T <- apply(abs(error_sim_std), 1, max)
T_Rep <- apply(abs(error_rep_std), 1, max)
```

(iv)

```r
plot(T_Rep ~ T, pch=".", cex=2)
abline(a=0,b=1)
```



(v)

```
mean(T_Rep >= T)
```

```
## [1] 0
```

p-value is 0 and this is the strong evidence for an outlier.

(vi)

The most extreme outlier is,

```
moores_df[unique(apply(abs(error_sim_std), 1, which.max)),]
```

```
##    Processor Year Transistors
## 49 ARM 9TDMI 1999      111000
```

which is "ARM 9TDMI" produced in 1999 that only has `111,000` transistors. In this time period, other processors have tens of millions of transistors.