Yaran Fan & Lei Li

Travel Time to Work in Cities:

Chicago, New York, Philadelphia & San Fransisco

Purpose

- After graduation, lots of students would work in big cities .
- No matter what industry a person is in, an important daily routine that could not be avoided by any employee is traveling to work.
- We would like to compare 4 major cities in the United States, to see what's the time that people spend on their way to work. Are there any differences or similarities? Or which city performs better from this perspective?

- According to the Census Bureau:
- “Travel time to work refers to the total number of minutes that it usually took the person to get from home to work each day during the reference week. The elapsed time includes time spent waiting for public transportation, picking up passengers in carpools, and time spent in other activities related to getting to work.”

Source: http://quickfacts.census.gov/qfd/meta/long_LFE305200.htm

Structure and Procedure

- Data collection and cleaning
- Exploratory Data Analysis (EDA)
- The Model – Hierarchical Regression(normal)
- The MCMC – Gibb Sampling
- Results and Checks
- Some Predictions
- Conclusion and Discussion
- Improvements

Data Collection and Cleaning

- Source: IPUMS-USA (http://usa.ipums.org/usa/)
- Survey data; 2010
- Variables:
  - Living City
  - Personal Weight
  - Number of Children under 5 years old
  - Age
  - Sex,
  - Employment Status
  - Class of Work
  - Personal Income
  - Family Income
  - Working City
  - Transportation
  - Travel Time
  - Departure Time
  - Arrival Time

Data Collection and Cleaning (cont.)

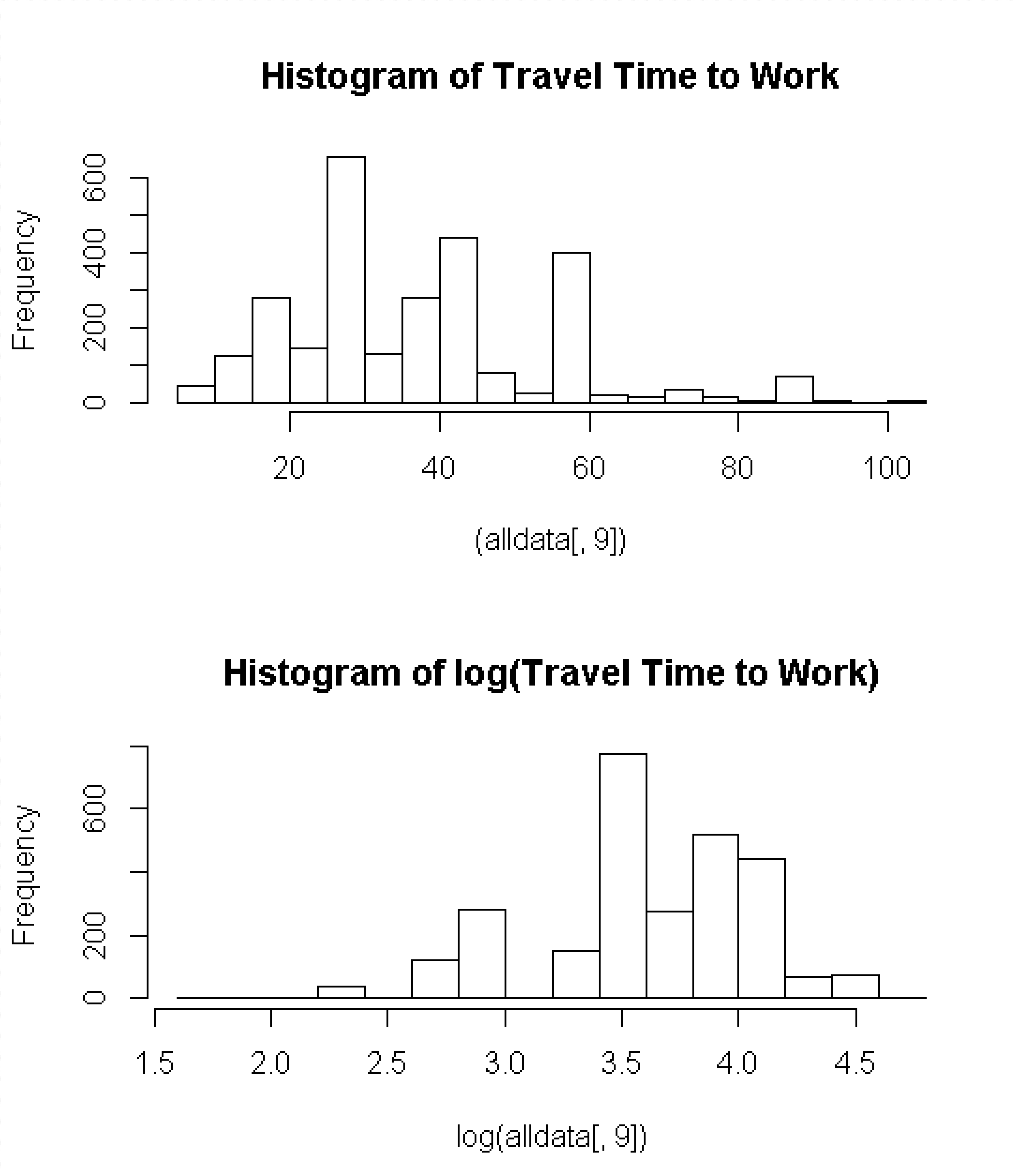
» Raw data: 3,061,692 Observations

» Conditions:

- ❖ Living and working in the same city(one of the four);
- ❖ Employed and not self-employed
- ❖ Traveling time is greater than 0 and less than 120 minutes
- ❖ Traveling by one of the 4 public transportation methods: bus, subway, taxi and railway
- ❖ Traveling in the morning hours: leave between 7:00 a.m. and 9:30 a.m.; arrive between 7:30 a.m. and 10:00 a.m.
- ❖ Keep all data of Philadelphia and San Francisco, whose sample sizes <1,000
- ❖ Randomly select 1,000 rows for New York (7,035 rows) and Chicago (1,022 rows)

» Clear data: 2,923 observations

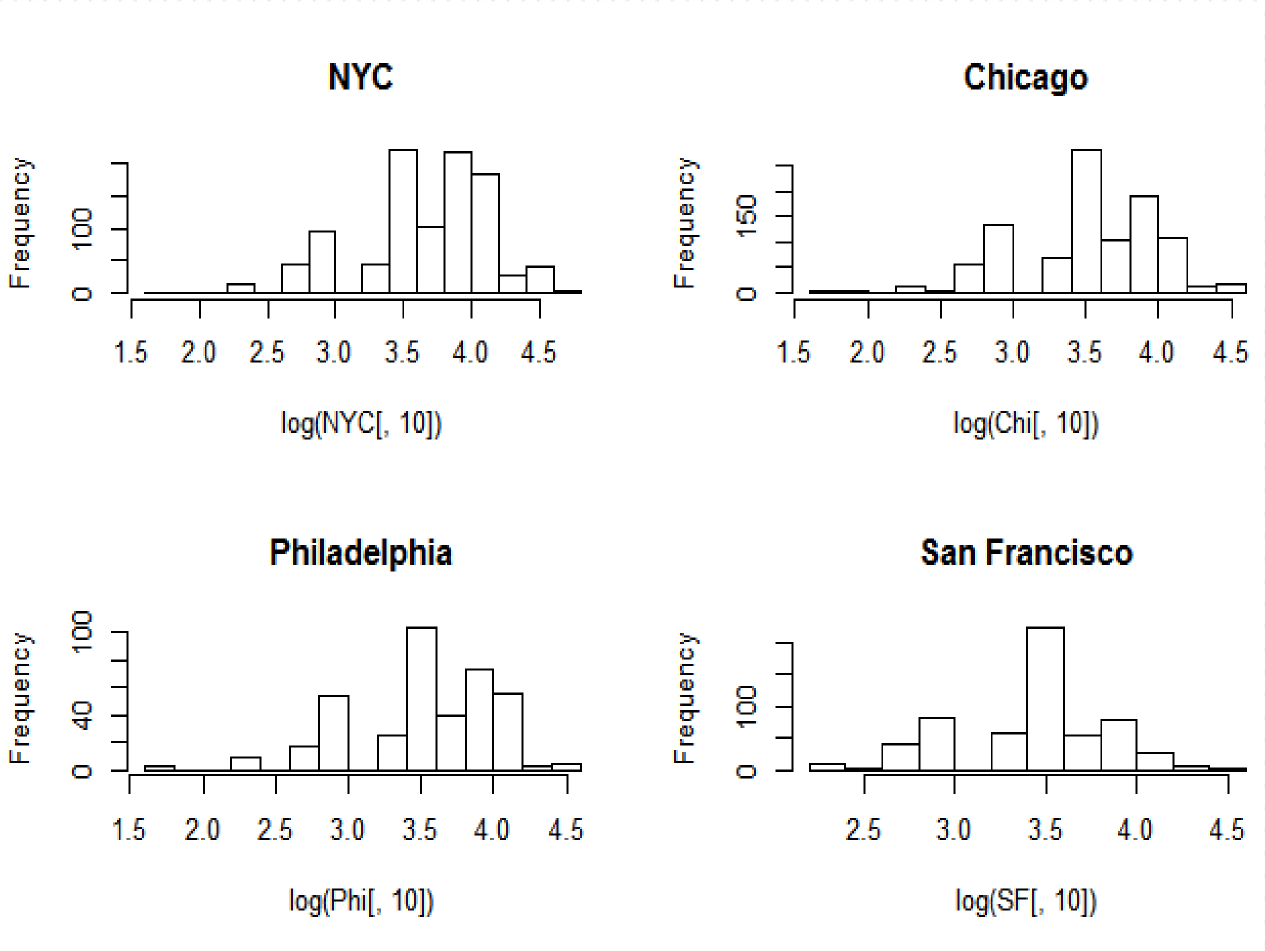
Exploratory Data Analysis (EDA)



Histograms – Total

- Peaks around 30, 60 and 90 minutes
- It is very possible that in such a survey, people tend to report the time rounded to a 30 minutes.
- If this is true, the peaks should be lower while the bars on both sides of the peaks would be higher.
- Then both plots will be closer to normal.
- So we assume our prior is a normal distribution.

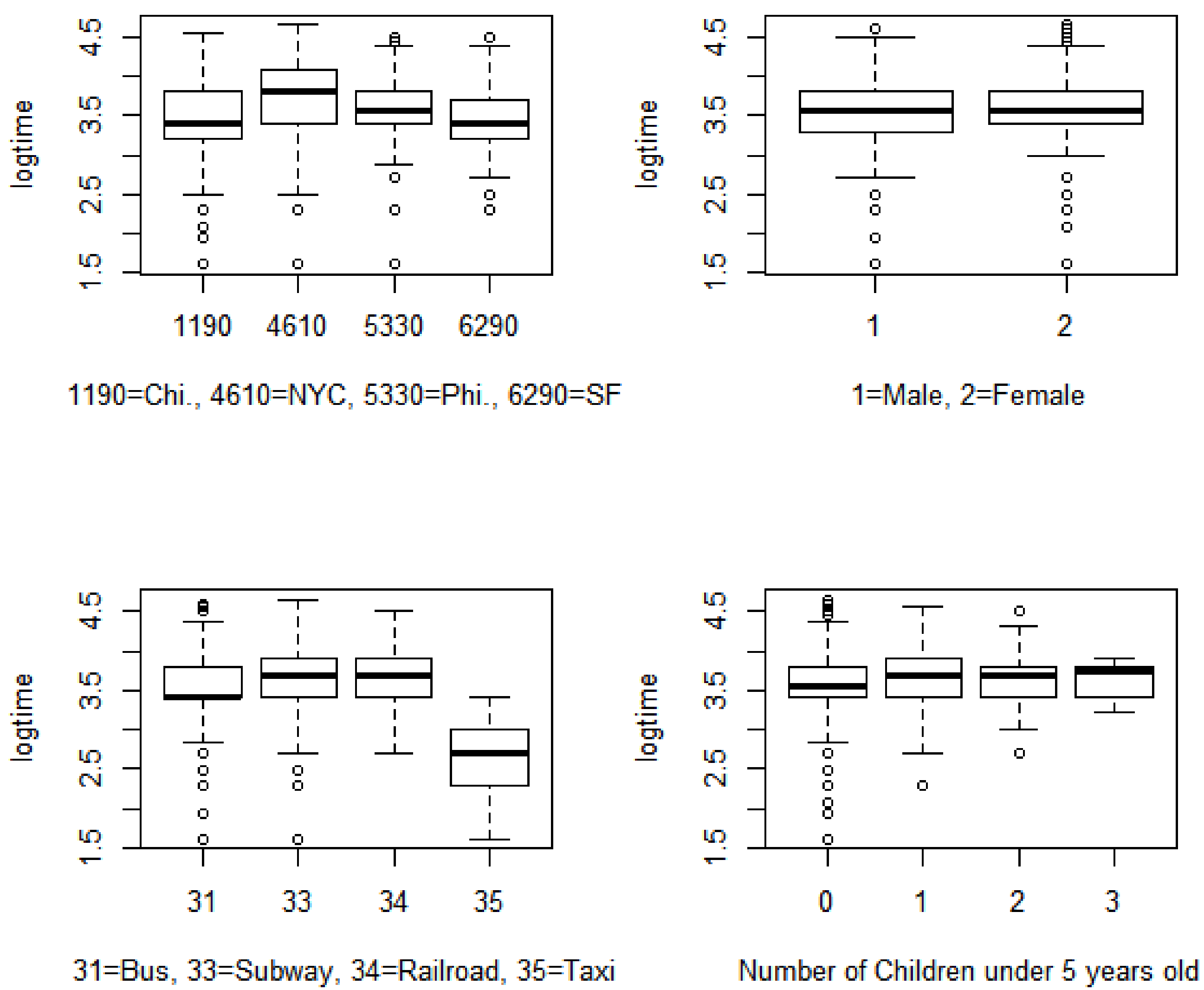
Exploratory Data Analysis (cont.)



Histograms –Cities (log)

- All four are similar to the second histogram on the last slide.
- They share quite similar pattern and range.
- We think the four cities could be classified as the same group but with different variations.

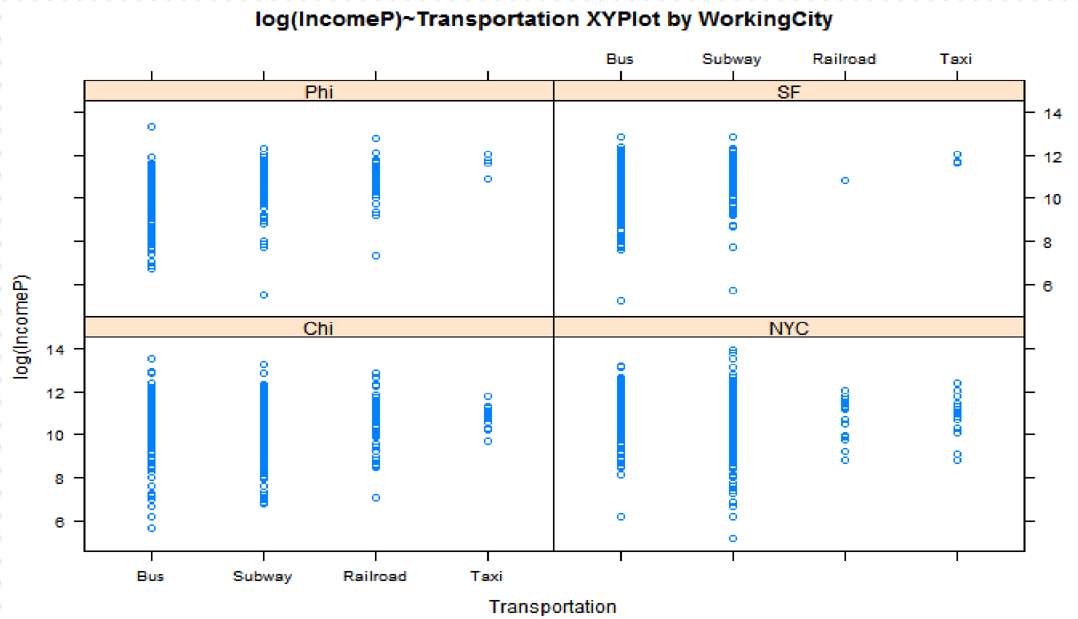
Exploratory Data Analysis (cont.)

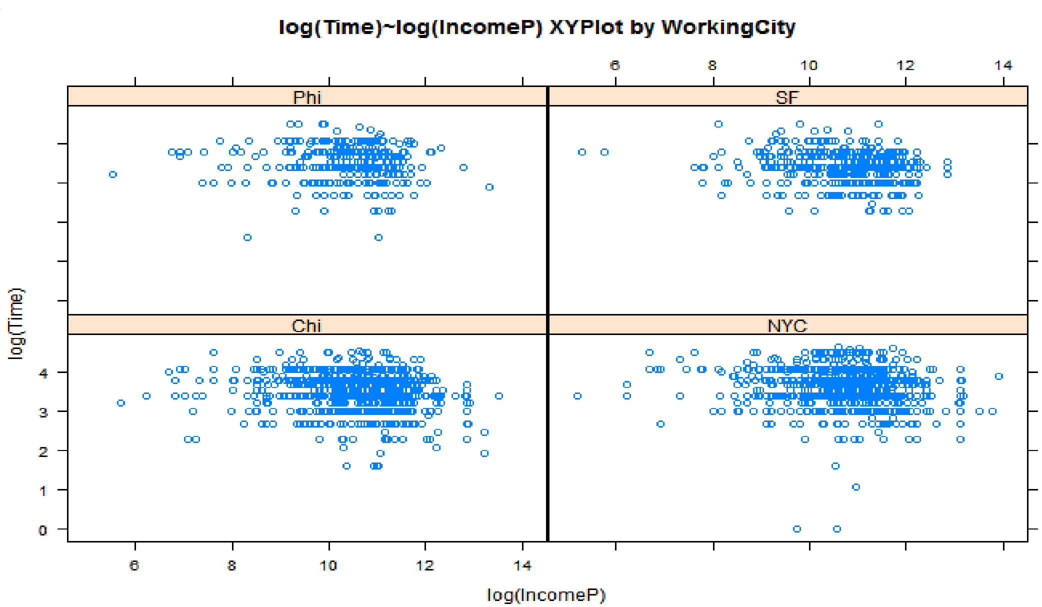


Box Plots

- Generally, the spreads are quite similar across the 4 cities.
- It seems female spend more time on their way to work.
- We did not do the analysis of railroad in our final results, because it seems railroad is used more often for inter-city traveling.
- Time goes up as children number increases.

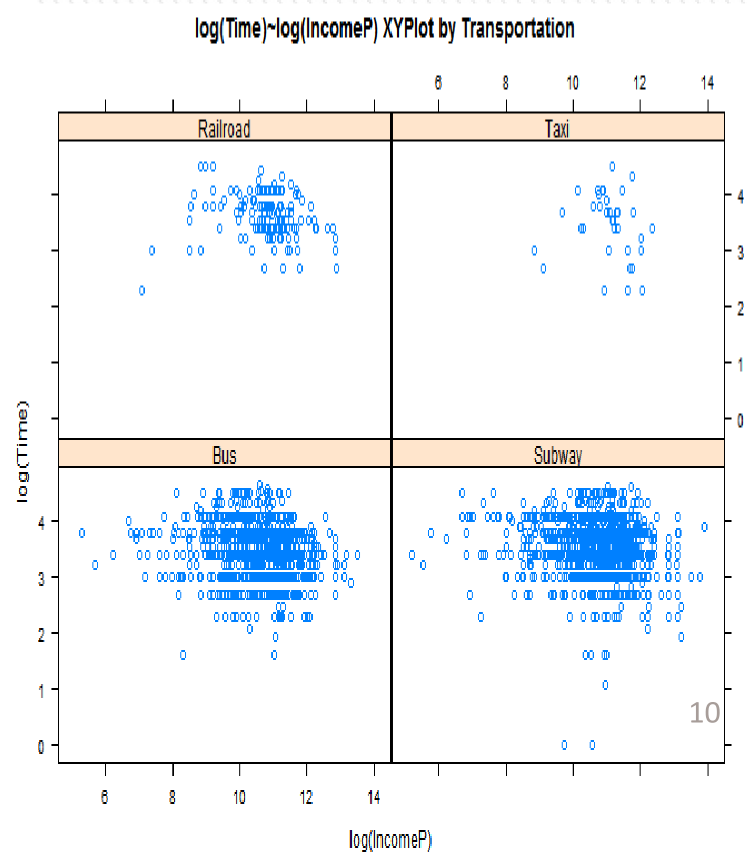
Exploratory Data Analysis (cont.)





XYPlots

The linear relations are not very obvious between traveling time and income



The Model

» According to our believes and explanations provided in the previous session, we propose the general model a Hierarchical Normal Regression Model. The four groups are the focusing cities and we assume they have different variances in the distribution.

» We use log transformation to make the data closer to normal and use centered age to represent the effects on the average age.

» The general regression model is:

$$\log(Time)_{ij} = \beta_{1j} + \beta_{2j} \times Children_{ij} + \beta_{3j} \times age_{ij} + \beta_{4j} \times Sex_{ij}$$
$$+\beta_{5j} \times \log(PersonalIncome_{ij}) + \beta_{6j} \times Bus_{ij}$$
$$+\beta_{7j} \times Subway_{ij} + \beta_{8j} \times Taxi_{ij} + \varepsilon_{ij}$$

Note: $age_{ij} = Age_{ij} - \overline{Age}$

The Model (cont.)

» Our prior believes are:

For the 4 cities:

$$\beta_1, \beta_2, \beta_3, \beta_4 \sim mvtNormal(\theta, \Sigma)$$

$$\theta \sim mvtNormal(\mu_0, \Lambda_0)$$

$$\Sigma \sim inverse-Wishart(\eta_0, S_0^{-1})$$

$$\varepsilon_{ij} \sim Normal(0, \sigma_j^2)$$

$$\sigma_j^2 \sim inverse-Gamma(\nu_0 / 2, \nu_0 \sigma_0^2 / 2)$$

$$\sigma_0^2 \sim Gamma(a, b)$$

MCMC – Gibb Sampling

» To do the MCMC (gibb sampling), we need the **full conditional distributions:**

» For $\beta_j\,(j=1,2,3,4)$

$$\beta_j \mid y_j, X_j, \sigma_j^2, \theta, \Sigma \sim mvtNormal(E[\beta_j], Var[\beta_j])$$

$$E[\beta_j \mid y_j, X_j, \sigma_j^2, \theta, \Sigma] = (\Sigma^{-1} + X_j^T X_j / \sigma_j^2)^{-1}(\Sigma^{-1}\theta + X_j^T y_j / \sigma_j^2)$$

$$Var[\beta_j \mid y_j, X_j, \sigma_j^2, \theta, \Sigma] = (\Sigma^{-1} + X_j^T X_j / \sigma_j^2)^{-1}$$

$$y_j = (\log Time_{1j}, \ldots, \log Time_{n_j j})^T$$

$$X_j = (1, Children, age, Sex, \log(PersonalIncome), Bus, Subway, Taxi)$$

MCMC – Gibb Sampling (cont.)

» For θ

$$\theta \mid \beta_1, \ldots, \beta_m, \Sigma \sim mvtNormal(\mu_m, \Lambda_m)$$

$$\Lambda_m = (\Lambda_0^{-1} + m\Sigma^{-1})^{-1}$$

$$\mu_m = \Lambda_m(\Lambda_0^{-1}\mu_0 + m\Sigma^{-1}\overline{\beta})$$

$$m = 4$$

» For Σ

$$\Sigma \mid \theta, \beta_1, \ldots, \beta_m \sim inverse-Wishart(\eta_0 + m, [S_0 + S_\theta]^{-1})$$

$$S_\theta = \sum_{j=1}^{m} (\beta_j - \theta)(\beta_j - \theta)^T$$

MCMC – Gibb Sampling (cont.)

» For σ_0^2

$$\sigma_0^2 \mid \upsilon_0, \sigma_j^2, m, a, b \sim Gamma(m\sigma_0^2 / 2 + a, \upsilon_0 / 2\sum_{j=1}^{m} 1 / \sigma_j^2 + b)$$

» For σ_j^2

$$\sigma_j^2 \mid \upsilon_0, \sigma_0^2, \beta_j, y_{i,j}, X_{i,j} \sim$$

$$Inverse - Gamma(\frac{\upsilon_0 + n_j}{2}, \frac{\upsilon_0\sigma_0^2 + \sum_{i=1}^{n_j} (y_{i,j} - \beta_j^T X_{i,j})^2}{2})$$

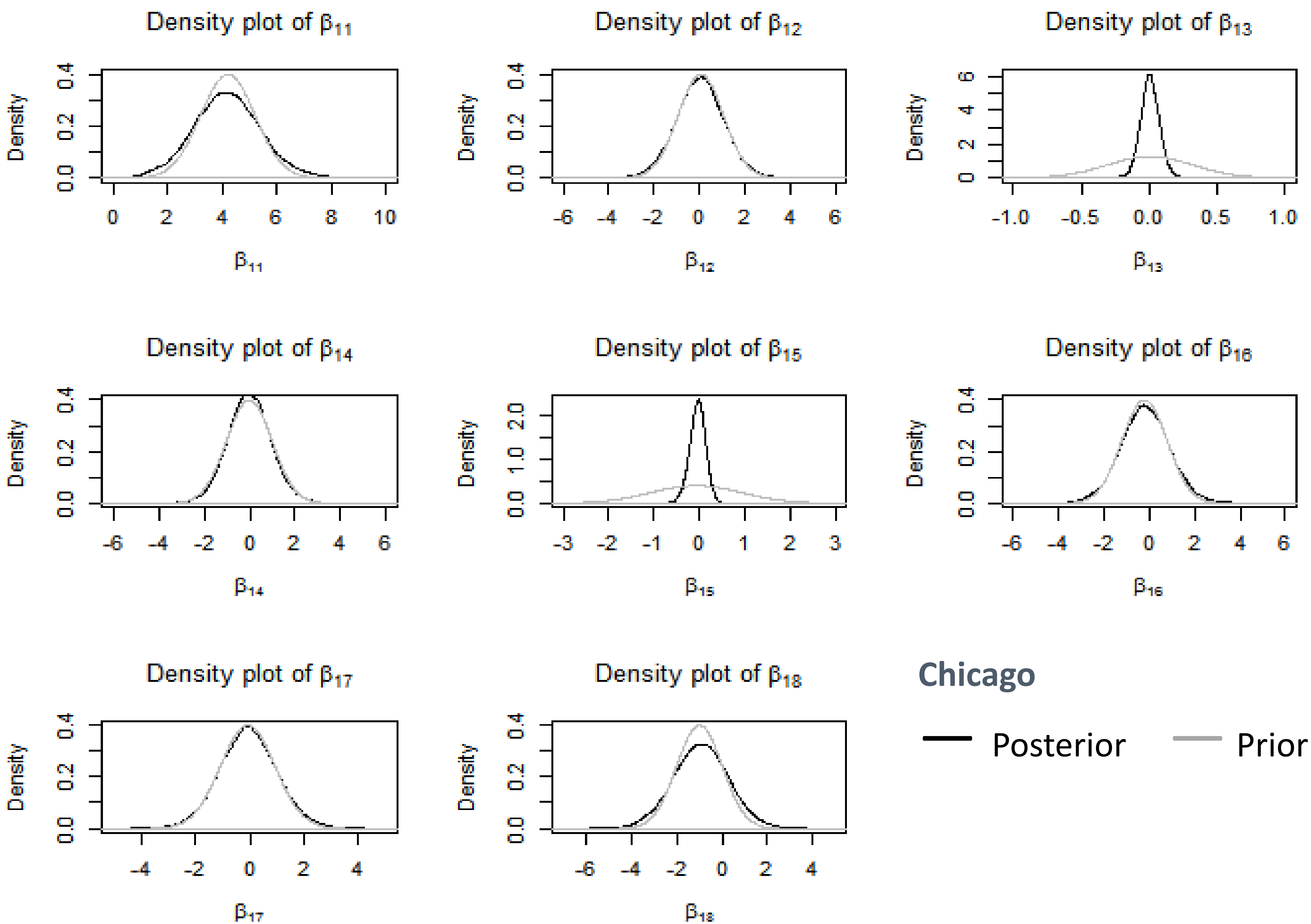
Results and Checks

As the posterior plots of most betas are around zero, we'd like to have a look at the mean before we look into all the posteriors.

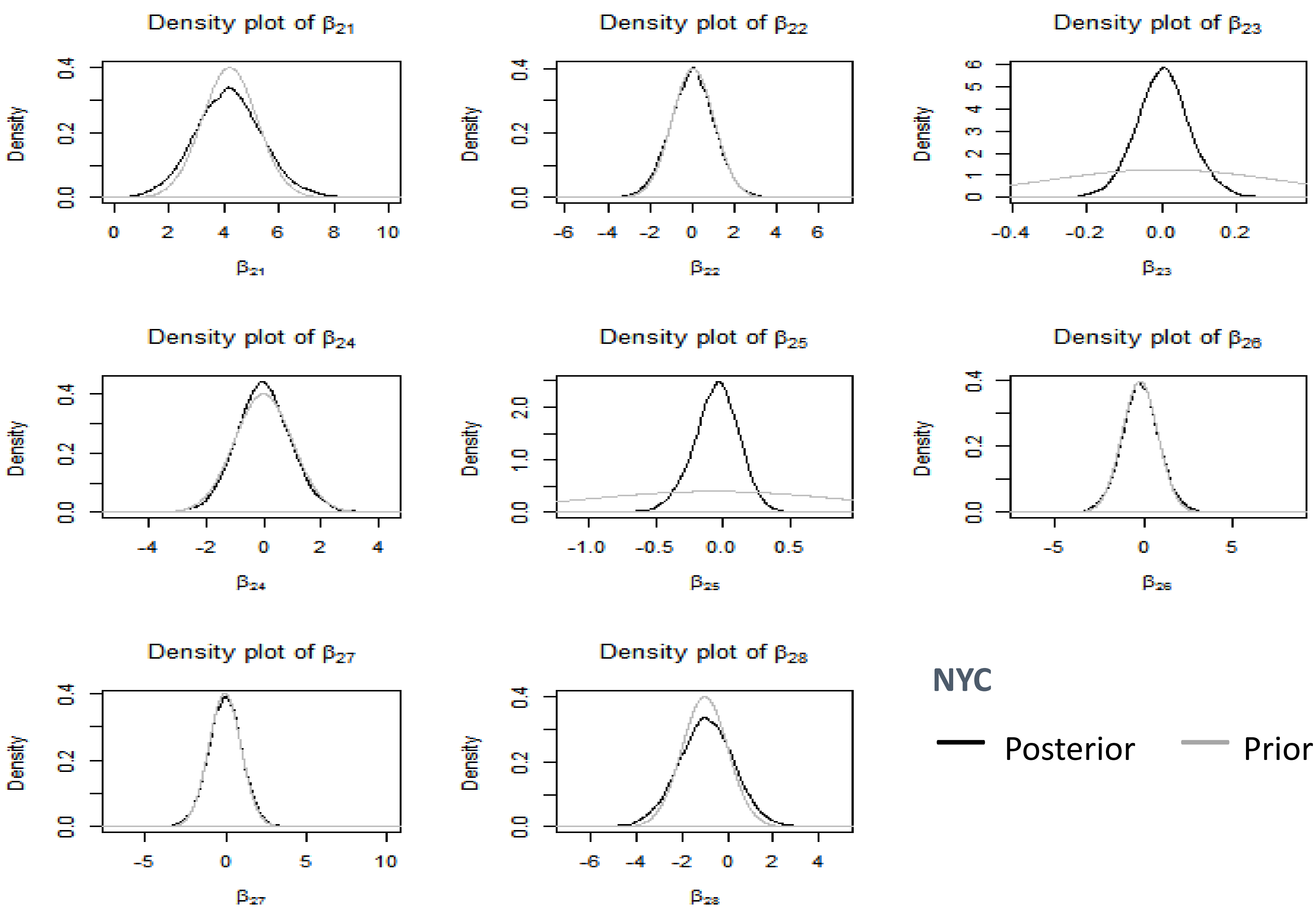
The means for all betas (also compared with OLS):

Beta	1	2	3	4	5	6	7	8
Bayesian Mean	4.211	0.049	0.004	-0.027	-0.050	-0.153	-0.088	-1.030
OLS	4.396	0.062	0.004	-0.013	-0.060	-0.210	-0.077	-0.993
Note	Constant	Children	Age	Sex	Personal Income	Bus	Subway	Taxi

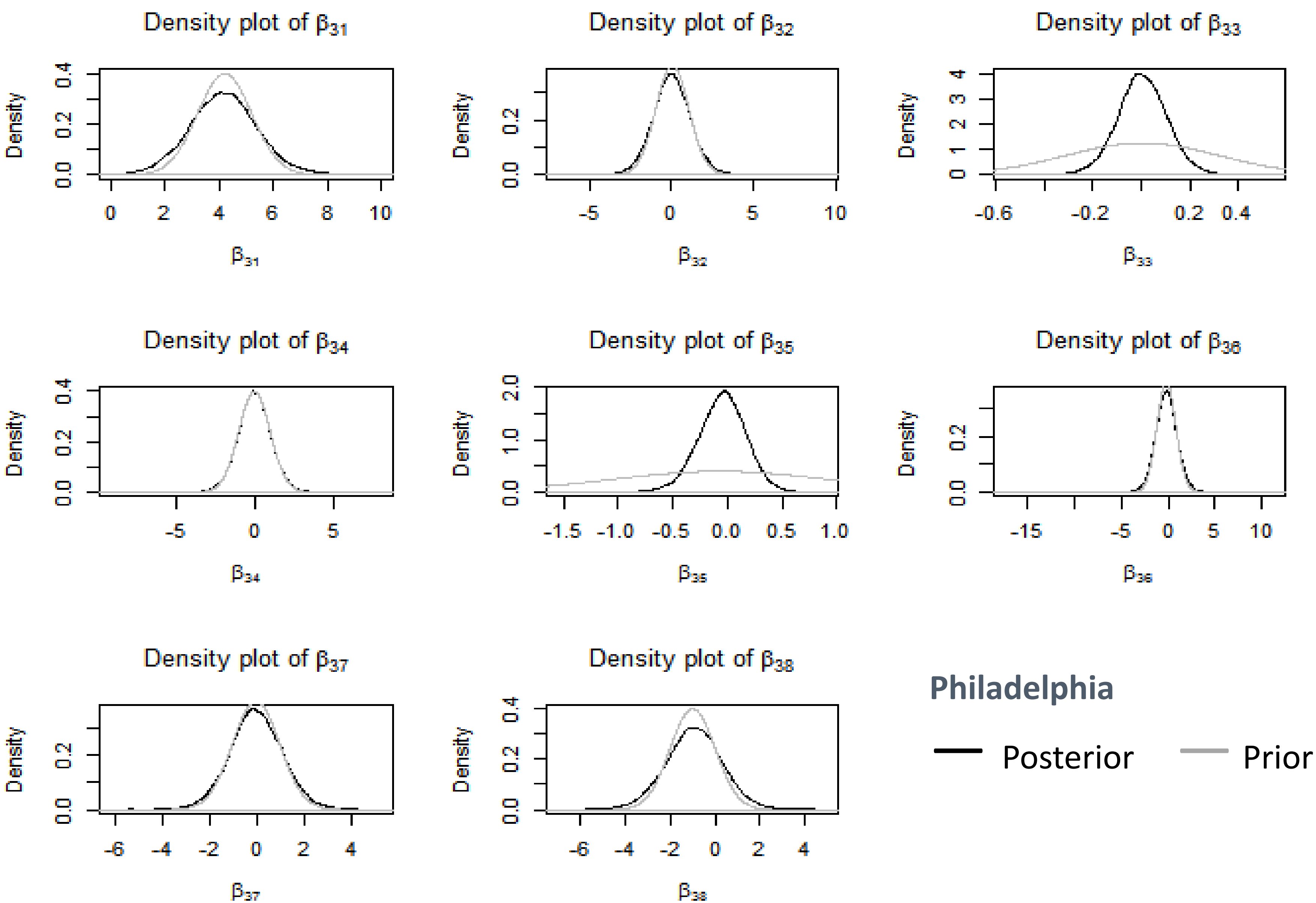
Results and Checks (cont.)



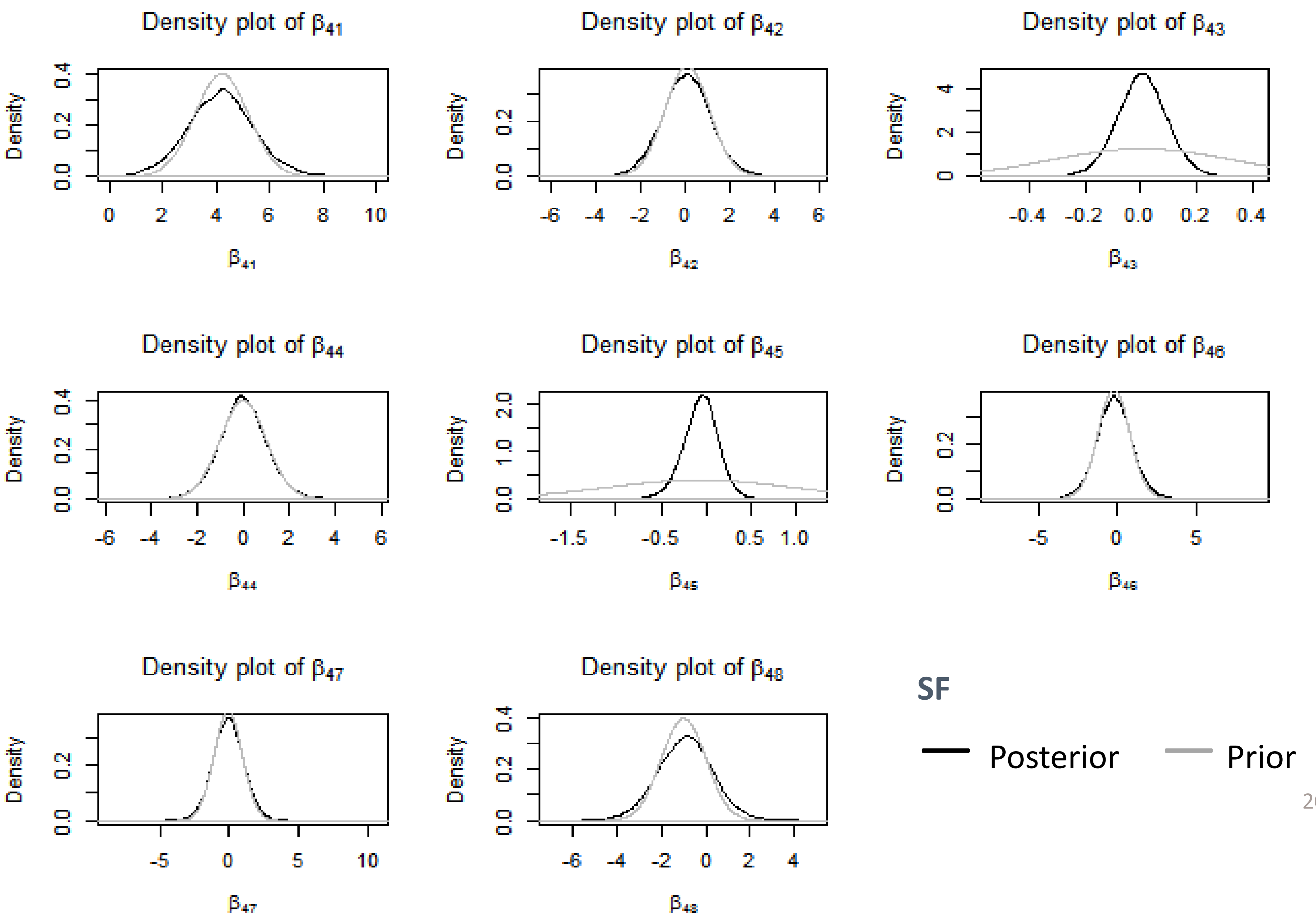
Results and Checks (cont.)



Results and Checks (cont.)



Results and Checks (cont.)



Results and Checks (cont.)

Results and Checks (cont.)

» **Auto correlation checks**

Results and Checks (cont.)

- **Gelman-Rubin Diagnostic:**
- All results are close to 1, which show convergences. (all results are attached)

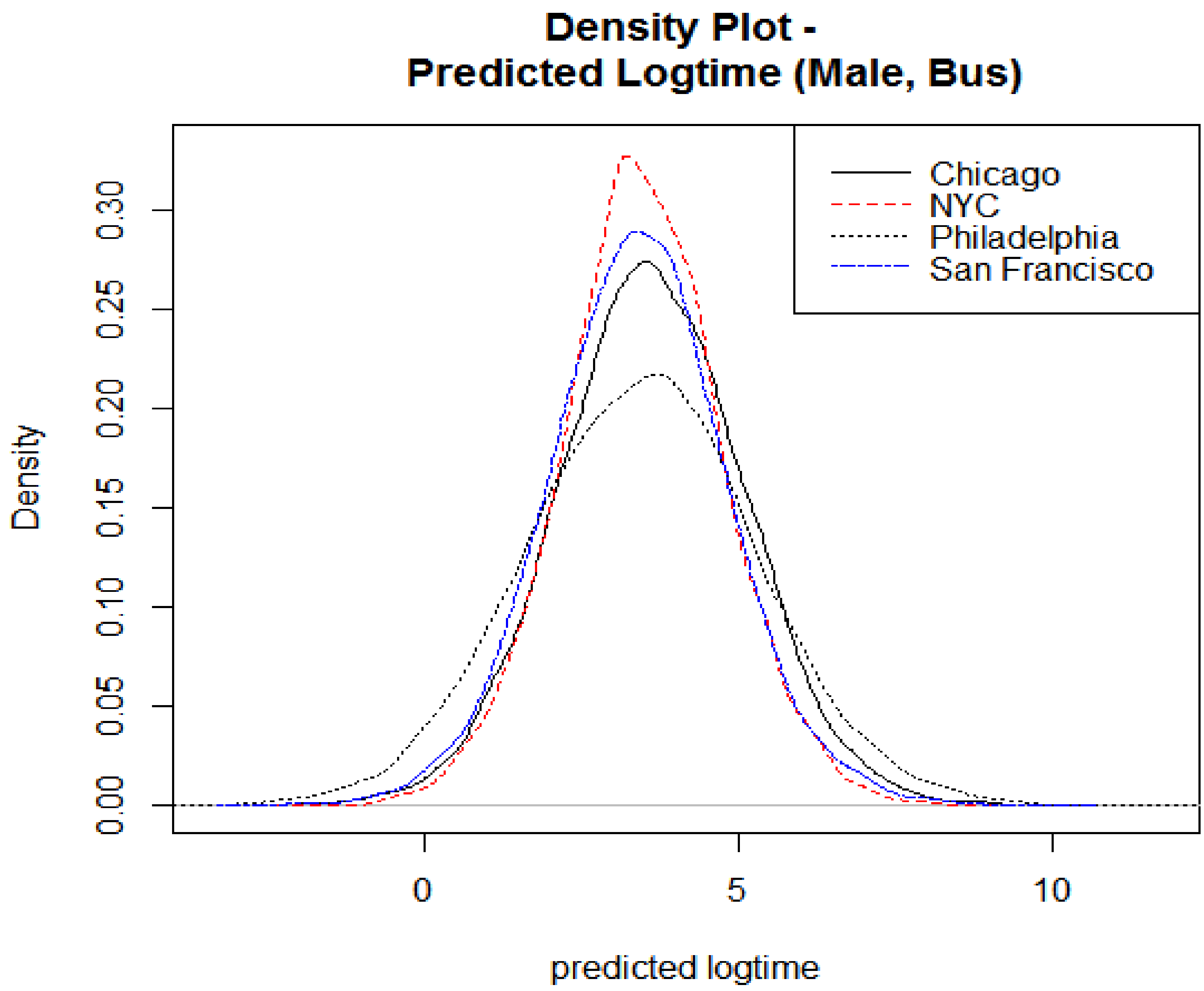
Eg:

```
> gelman.diag(GR.BETA11)
Potential scale reduction factors:
     Point est. Upper C.I.
[1,]          1       1.02
```

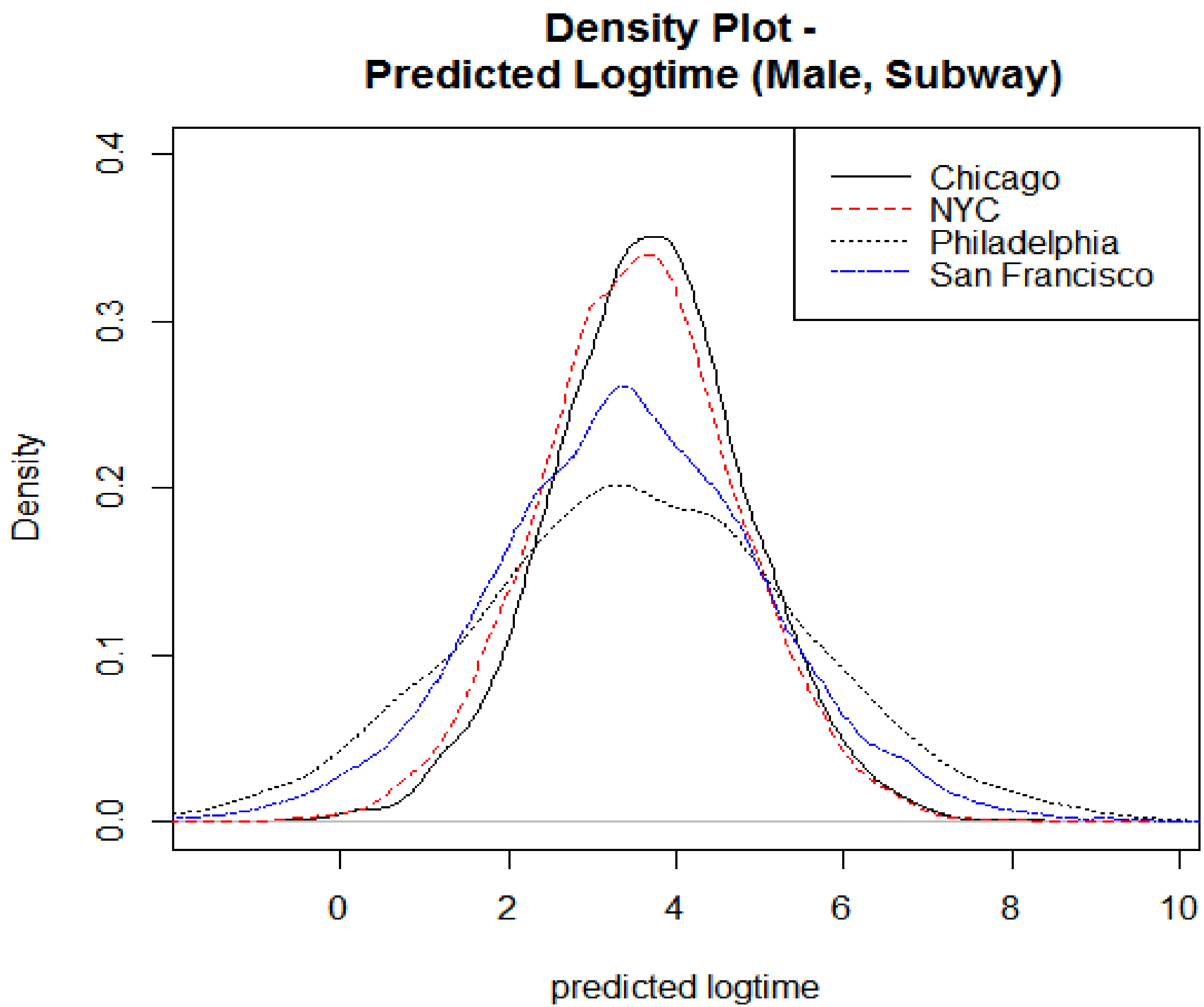
Eg:

```
> gelman.diag(GR.BETA48)
Potential scale reduction factors:
     Point est. Upper C.I.
[1,]          1          1
```

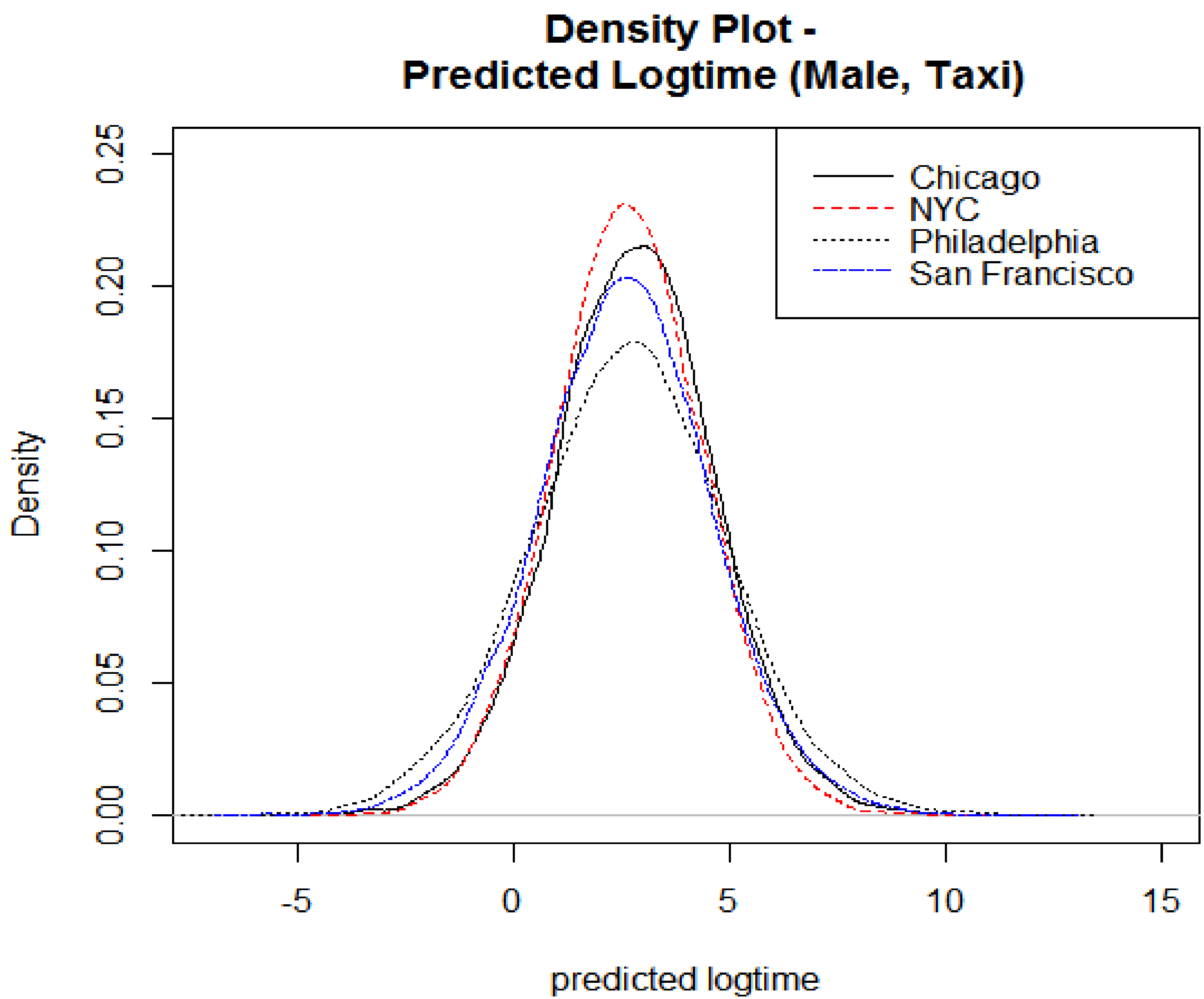
Some Predictions



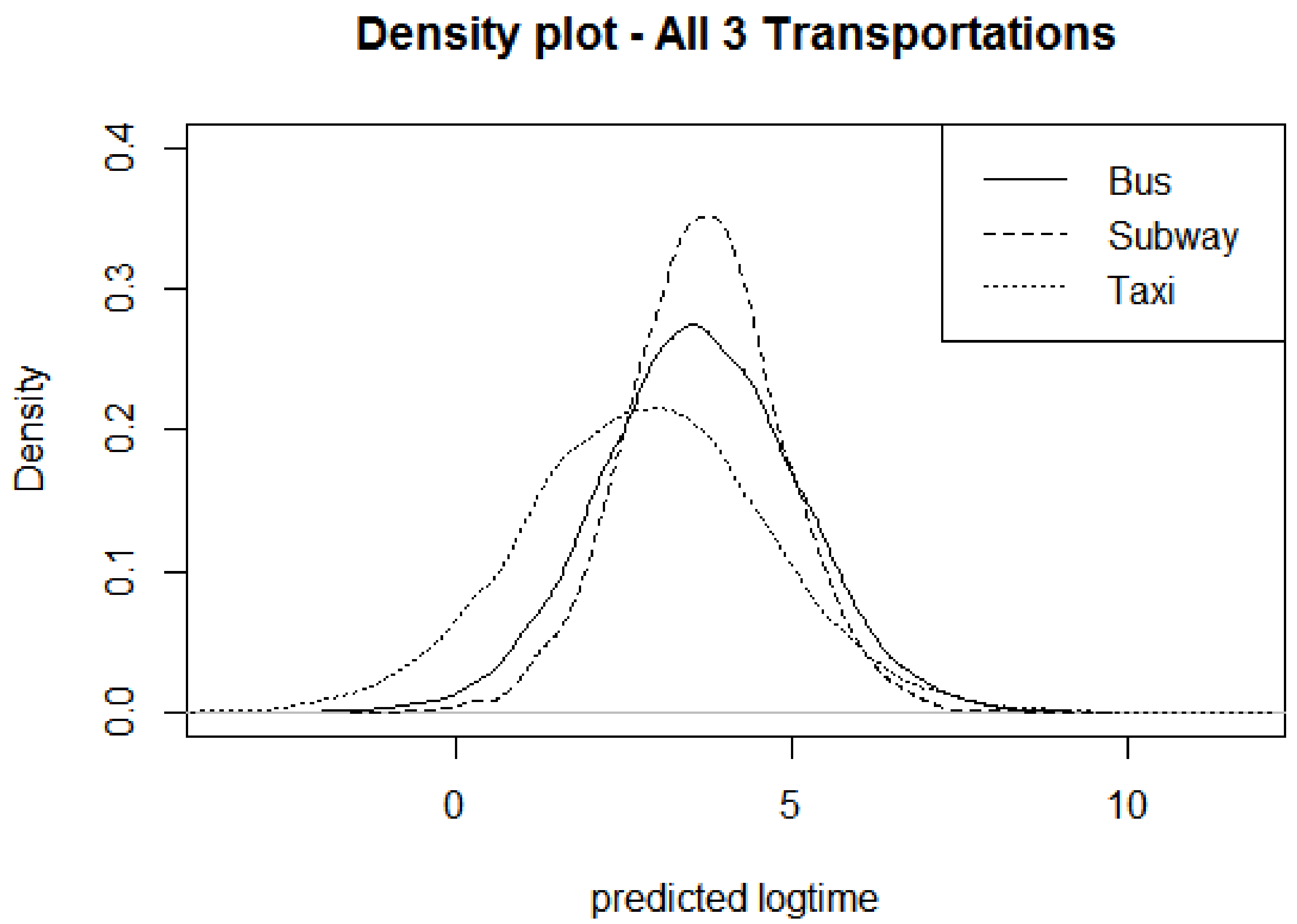
Some Predictions (cont.)



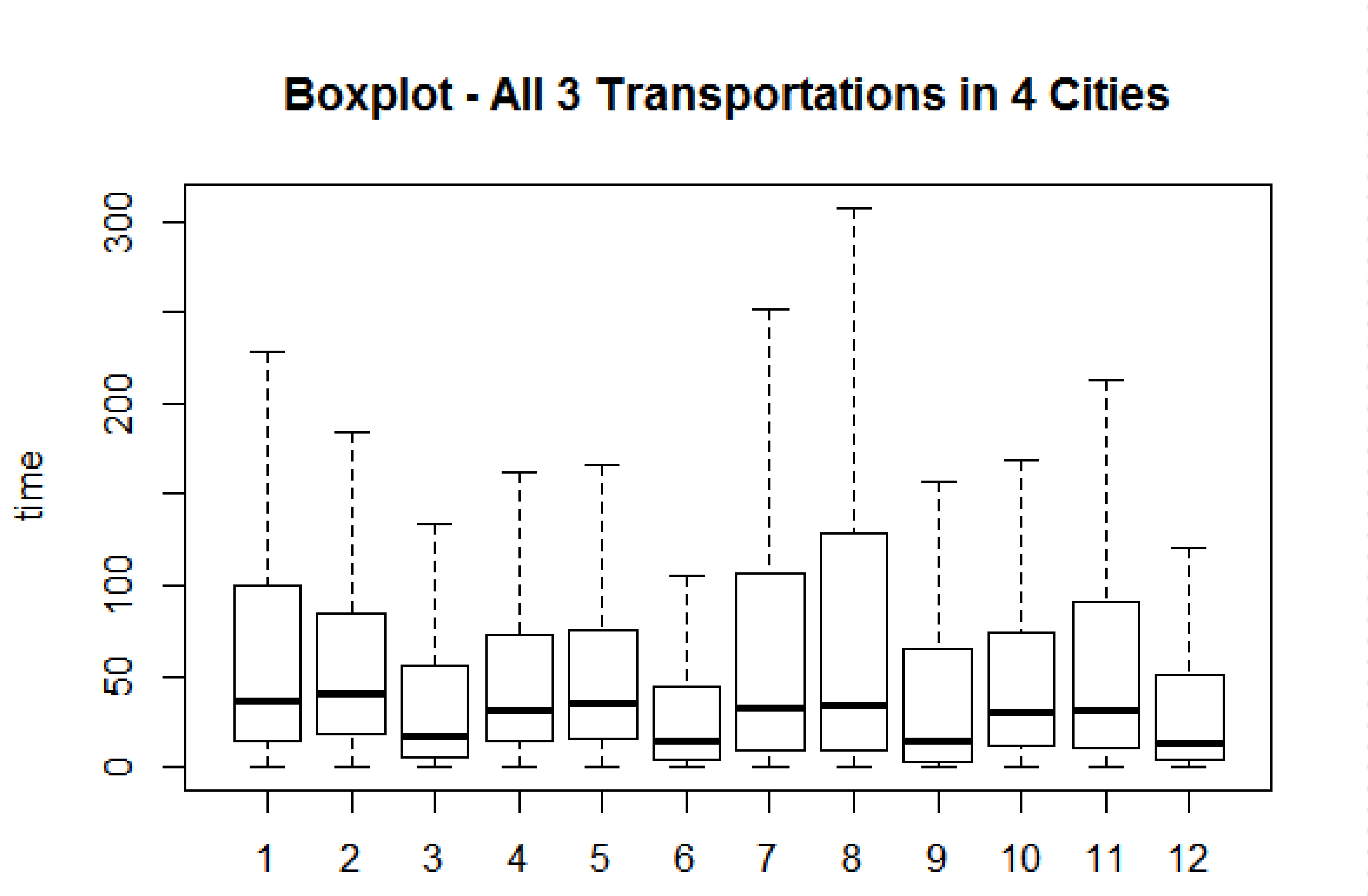
Some Predictions (cont.)



Some Predictions (cont.)



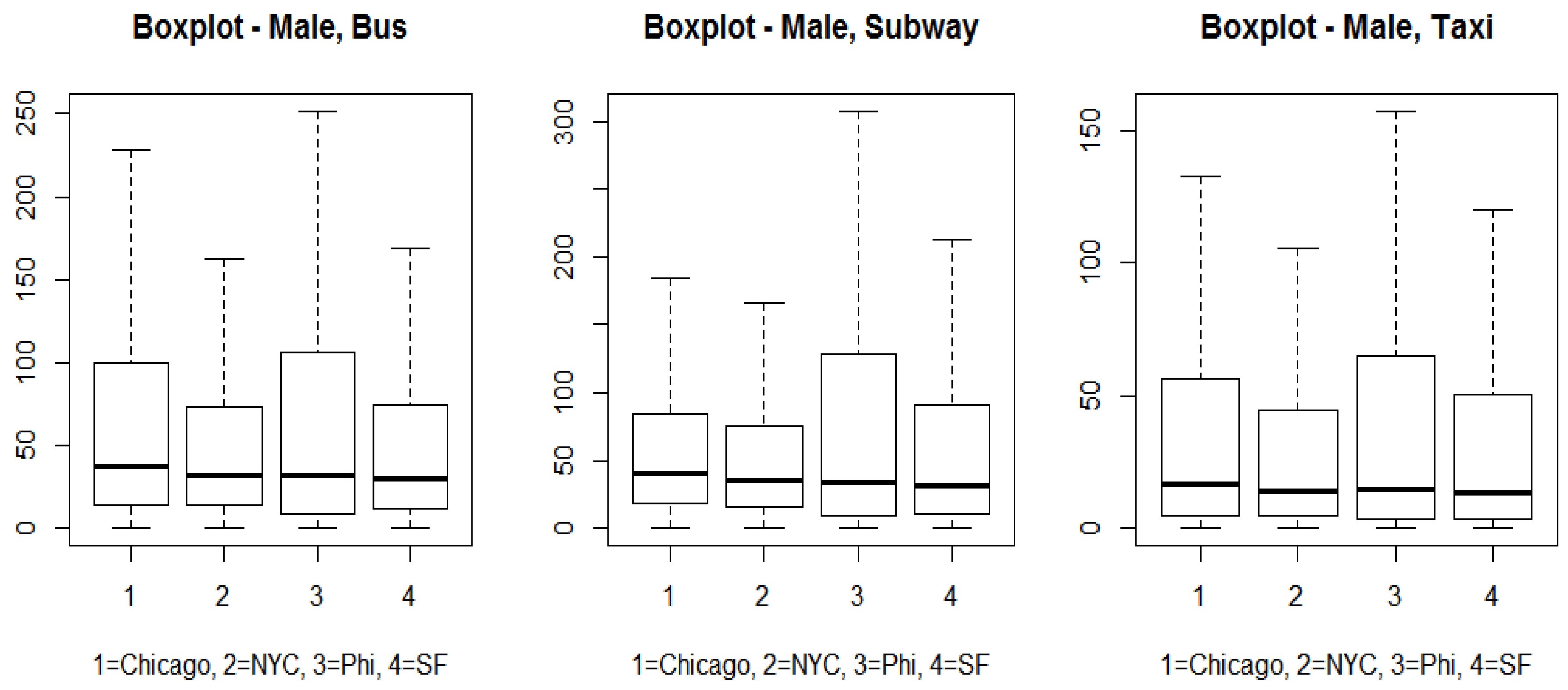
Some Predictions (cont.)



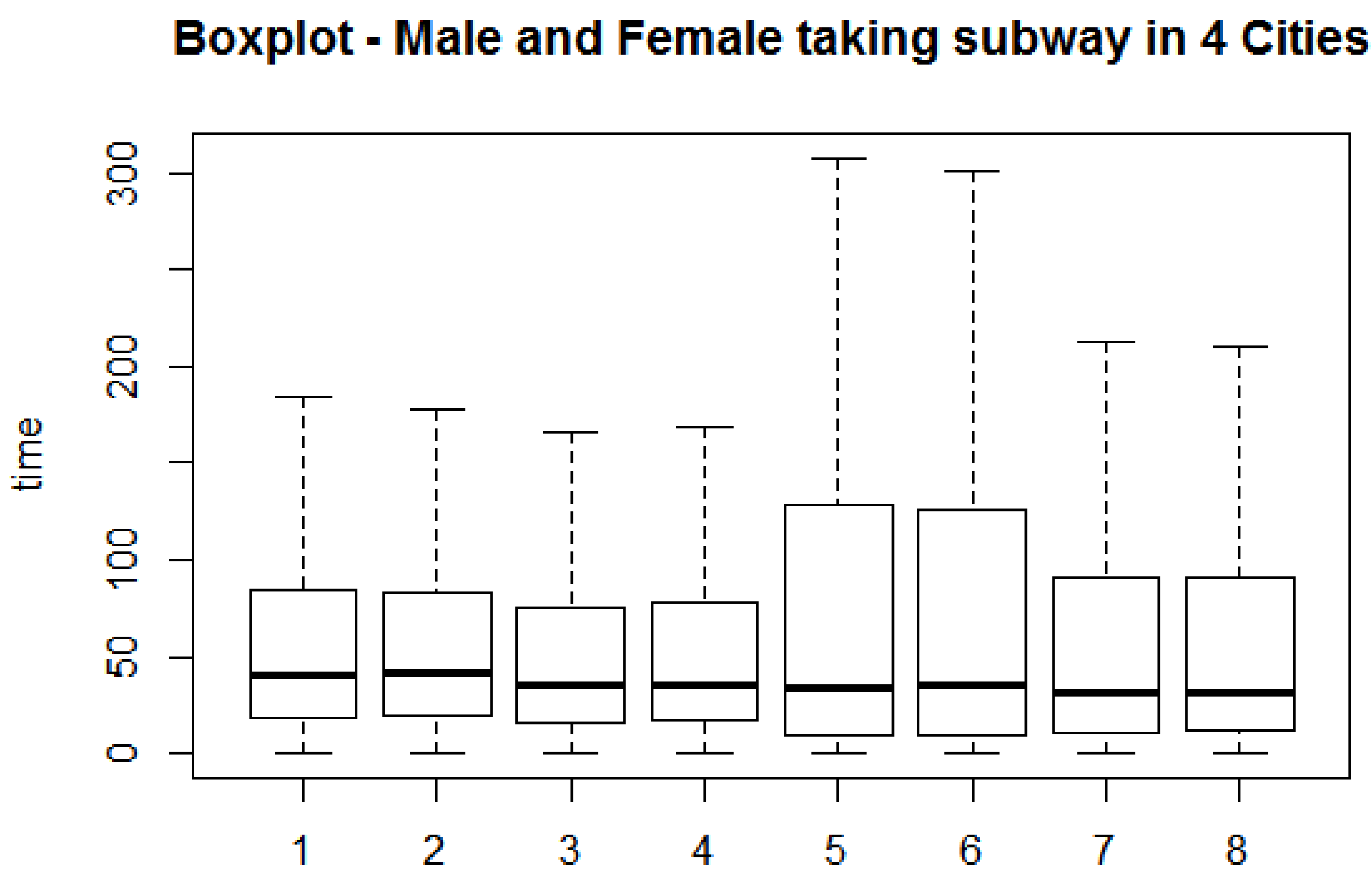
(male without children)

#1 Chi bus #2 Chi subway #3 Chi taxi #4 NYC bus #5 NYC subway #6 NYC taxi
#7 Phi bus #8 Phi subway #9 Phi taxi #10 SF bus #11 Sfsubway #12 SF taxi

Some Predictions (cont.)

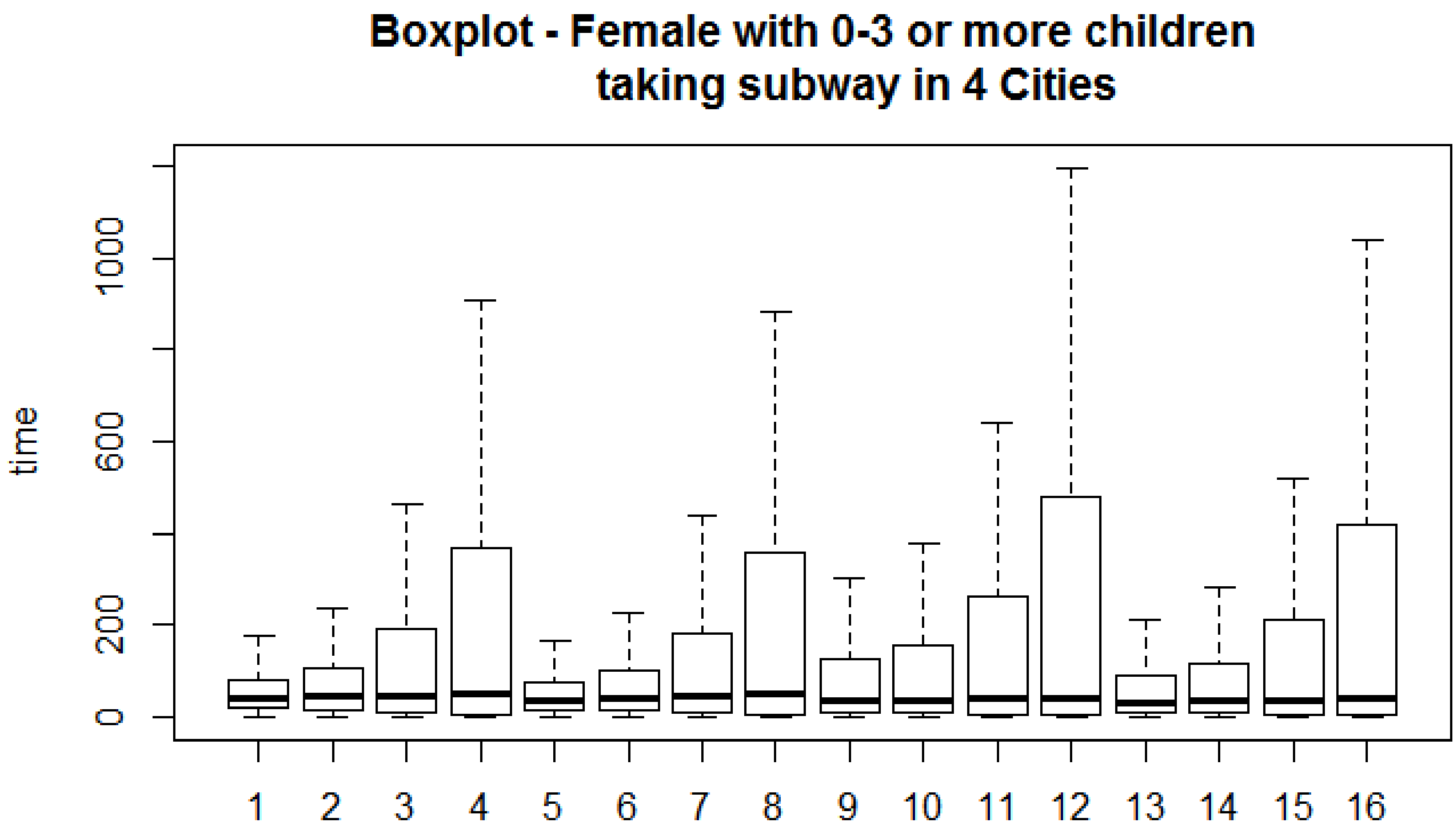


Some Predictions (cont.)



#1 Chi M #2 Chi F #3 NYC M #4 NYC F
#5 Phi M #6 Phi F #7 SF M #8 SF F

Some Predictions (cont.)



#1 Chi M   #2 Chi F   #3 NYC M   #4 NYC F

#1-4 Chi   #5-8 NYC   #9-12 Phi   # 13-16 SF

No Children: 1,5,9,13   1 Child: 2, 6,10,14

2 Children: 3, 7, 11,15   3 or More Children: 4, 8, 12, 16

Conclusion and Discussion

- 1. Generally, the posterior predictions are quite similar for all 4 cities. New York performs better in all 3 public transportation methods(male without children). Chicago and San Francisco seem to be moderate. Philadelphia does not perform as good as the others.
- 2. In general, subway is comparatively slow, and taxi is the fastest way among the 3 transportations. All three ways generally are faster than railroad.
- 3. The number of children (under 5 years old) of their own has a positive relation with the potential travel time to work.
- 4. From the beta's posterior density plot, it seems that age and personal income do not have obvious effects on travel time to work.

Conclusion and Discussion (cont.)

Discussions:

- 1. From the EDA, New York seems to have a higher log(time), however in the predicted (male no children) data, it performs better than the other 3 cities. One important factor is “with/without children”. Among the 2923 observations, only 261 are with children(one or more). And 114 of them are from New York. As a person with more children tends to have longer traveling time, this could contribute to the larger log(time) in the EDA. When predict using “without children”, New York’s performance improves.
- 2. Another reason for New York’s good performance regardless of the huge population is, the city adapted its public transportation system to the large population. The coverage and frequency of buses and subways could help make the travel faster in NYC than in the other four.
- 3. It’s reasonable that subway takes longer time as it often covers the largest area compared with bus and taxi. Taxi takes the shortest time as it is usually for short distance trip and there are no stops during the trip.

Improvements

- 1. The distribution of the travel time is not easy to identify as there are gaps in it. Instead of normal models, we could try other distributions that may fit the data better.

- 2. The study here only focus on public transportations. However, it could not represent the exact travel time to work of the whole population. One of the reasons of Chicago, Philadelphia and San Francisco’s less satisfactory performance could be that we did not take private transportations into consideration. Driving, carpooling and some other ways could be also considered in further studies. They would make the results more comprehensive.

» Appendix:

- A. Autocorrelation and Gelman-Rubin Results
  (page 36 – 40)

- B. R Codes
  (page41-58)

```
>
autocorr(as.mcmc(BETA.post[1,1,1
0000:20000]))
, , 1

              [,1]
Lag 0   1.00000000
Lag 1   0.66719298
Lag 5   0.47638332
Lag 10  0.32453244
Lag 50 -0.02870706

>
autocorr(as.mcmc(BETA.post[1,2,1
0000:20000]))
, , 1

             [,1]
Lag 0  1.00000000
Lag 1  0.56355000
Lag 5  0.34725596
Lag 10 0.19605407
Lag 50 0.02466666

>
autocorr(as.mcmc(BETA.post[1,3,1
0000:20000]))
, , 1

               [,1]
Lag 0   1.000000000
Lag 1   0.062830021
Lag 5  -0.004175722
Lag 10  0.007246290
Lag 50 -0.014305205

>
autocorr(as.mcmc(BETA.post[1,4,1
0000:20000]))
, , 1

              [,1]
Lag 0   1.00000000
Lag 1   0.56161124
Lag 5   0.31327115
Lag 10  0.15662916
Lag 50 -0.03885472

>
autocorr(as.mcmc(BETA.post[1,5,1
0000:20000]))
, , 1

               [,1]
Lag 0   1.000000000
Lag 1   0.451885793
Lag 5   0.301540320
Lag 10  0.216794618
Lag 50 -0.005272938

>
autocorr(as.mcmc(BETA.post[1,6,1
0000:20000]))
, , 1

                [,1]
Lag 0   1.0000000000
Lag 1   0.5821394875
Lag 5   0.3434217211
Lag 10  0.1888836695
Lag 50 -0.0002186341

>
autocorr(as.mcmc(BETA.post[1,7,1
0000:20000]))
, , 1

             [,1]
Lag 0  1.00000000
Lag 1  0.58001616
Lag 5  0.37144801
Lag 10 0.24329007
Lag 50 0.00259497

>
autocorr(as.mcmc(BETA.post[1,8,1
0000:20000]))
, , 1

             [,1]
Lag 0  1.00000000
Lag 1  0.65441616
Lag 5  0.45031055
Lag 10 0.30410991
Lag 50 0.01634156

>
autocorr(as.mcmc(BETA.post[2,1,1
0000:20000]))
, , 1

              [,1]
Lag 0   1.00000000
Lag 1   0.67736977
Lag 5   0.47165995
Lag 10  0.32387232
Lag 50 -0.02736504

>
autocorr(as.mcmc(BETA.post[2,2,1
0000:20000]))
, , 1

             [,1]
Lag 0  1.00000000
Lag 1  0.57935866
Lag 5  0.35939969
Lag 10 0.21245279
Lag 50 0.04771707

>
autocorr(as.mcmc(BETA.post[2,3,1
0000:20000]))
, , 1

                [,1]
Lag 0   1.0000000000
Lag 1   0.0727178266
```

```
Lag 5  -0.0001420175
Lag 10 -0.0028719962
Lag 50  0.0079636451

>
autocorr(as.mcmc(BETA.post[2,4,1
0000:20000]))
, , 1

              [,1]
Lag 0   1.00000000
Lag 1   0.55447826
Lag 5   0.31227900
Lag 10  0.13058939
Lag 50 -0.02921819

>
autocorr(as.mcmc(BETA.post[2,5,1
0000:20000]))
, , 1

              [,1]
Lag 0   1.00000000
Lag 1   0.46891684
Lag 5   0.32182173
Lag 10  0.21398971
Lag 50 -0.02143443

>
autocorr(as.mcmc(BETA.post[2,6,1
0000:20000]))
, , 1

             [,1]
Lag 0  1.00000000
Lag 1  0.59014220
Lag 5  0.35929318
Lag 10 0.18988728
Lag 50 0.00506986

>
autocorr(as.mcmc(BETA.post[2,7,1
0000:20000]))
, , 1

             [,1]
Lag 0  1.00000000
Lag 1  0.61382061
Lag 5  0.40429223
Lag 10 0.25723082
Lag 50 0.02206856

>
autocorr(as.mcmc(BETA.post[2,8,1
0000:20000]))
, , 1

             [,1]
Lag 0  1.00000000
Lag 1  0.64880065
Lag 5  0.46566930
Lag 10 0.31905088
Lag 50 0.02920156

>
autocorr(as.mcmc(BETA.post[3,1,1
0000:20000]))
, , 1

             [,1]
Lag 0   1.0000000
Lag 1   0.6317846
Lag 5   0.4337504
Lag 10  0.2894999
Lag 50 -0.0369534

>
autocorr(as.mcmc(BETA.post[3,2,1
0000:20000]))
, , 1

               [,1]
Lag 0  1.000000000
Lag 1  0.554941619
Lag 5  0.323388577
Lag 10 0.191548070
Lag 50 0.007099913

>
autocorr(as.mcmc(BETA.post[3,3,1
0000:20000]))
, , 1

                [,1]
Lag 0   1.000000000
Lag 1   0.107534415
Lag 5   0.015808971
Lag 10 -0.007958077
Lag 50  0.004293476

>
autocorr(as.mcmc(BETA.post[3,4,1
0000:20000]))
, , 1

             [,1]
Lag 0   1.0000000
Lag 1   0.5393573
Lag 5   0.2988819
Lag 10  0.1326215
Lag 50 -0.0242484

>
autocorr(as.mcmc(BETA.post[3,5,1
0000:20000]))
, , 1

                 [,1]
Lag 0   1.0000000000
Lag 1   0.2861944700
Lag 5   0.1876502682
Lag 10  0.1394365982
Lag 50 -0.0001914729

>
autocorr(as.mcmc(BETA.post[3,6,1
0000:20000]))
, , 1
```

```
              [,1]
Lag 0   1.0000000
Lag 1   0.5402055
Lag 5   0.3150021
Lag 10  0.1731689
Lag 50 -0.0165344

> 
autocorr(as.mcmc(BETA.post[3,7,1
0000:20000]))
, , 1

              [,1]
Lag 0  1.000000000
Lag 1  0.556457452
Lag 5  0.357550699
Lag 10 0.226244785
Lag 50 0.008574109

> 
autocorr(as.mcmc(BETA.post[3,8,1
0000:20000]))
, , 1

              [,1]
Lag 0  1.00000000
Lag 1  0.61764665
Lag 5  0.41875246
Lag 10 0.27582953
Lag 50 0.02905269

> 
autocorr(as.mcmc(BETA.post[4,1,1
0000:20000]))
, , 1

              [,1]
Lag 0   1.00000000
Lag 1   0.64634023
Lag 5   0.45669741
Lag 10  0.30943824
Lag 50 -0.03366028

> 
autocorr(as.mcmc(BETA.post[4,2,1
0000:20000]))
, , 1

              [,1]
Lag 0  1.00000000
Lag 1  0.57424357
Lag 5  0.33155193
Lag 10 0.18553543
Lag 50 0.00607718

> 
autocorr(as.mcmc(BETA.post[4,3,1
0000:20000]))
, , 1

              [,1]
Lag 0   1.000000000
Lag 1   0.097414091
Lag 5   0.015808969
Lag 10  0.003066199
Lag 50 -0.001211644

> 
autocorr(as.mcmc(BETA.post[4,4,1
0000:20000]))
, , 1

              [,1]
Lag 0   1.00000000
Lag 1   0.52052695
Lag 5   0.29417575
Lag 10  0.14012397
Lag 50 -0.02794255

> 
autocorr(as.mcmc(BETA.post[4,5,1
0000:20000]))
, , 1

              [,1]
Lag 0   1.00000000
Lag 1   0.38237053
Lag 5   0.25215851
Lag 10  0.17143141
Lag 50 -0.02076182

> 
autocorr(as.mcmc(BETA.post[4,6,1
0000:20000]))
, , 1

               [,1]
Lag 0   1.000000000
Lag 1   0.554642157
Lag 5   0.337003309
Lag 10  0.189608186
Lag 50 -0.006519239

> 
autocorr(as.mcmc(BETA.post[4,7,1
0000:20000]))
, , 1

               [,1]
Lag 0   1.000000000
Lag 1   0.567444529
Lag 5   0.351010000
Lag 10  0.229993782
Lag 50 -0.005889355

> 
autocorr(as.mcmc(BETA.post[4,8,1
0000:20000]))
, , 1

             [,1]
Lag 0  1.00000000
Lag 1  0.64372692
Lag 5  0.44503930
Lag 10 0.28543024
Lag 50 0.01198086
```

```
> gelman.diag(GR.BETA11)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.02

> gelman.diag(GR.BETA12)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA13)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA14)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.01

> gelman.diag(GR.BETA15)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA16)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.02

> gelman.diag(GR.BETA17)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.01

> gelman.diag(GR.BETA18)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.01

> gelman.diag(GR.BETA21)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.01

> gelman.diag(GR.BETA22)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA23)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA24)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA25)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA26)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.01

> gelman.diag(GR.BETA27)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.01

> gelman.diag(GR.BETA28)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA31)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.01

> gelman.diag(GR.BETA32)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1
```

```
> gelman.diag(GR.BETA33)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA34)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA35)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA36)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.01

> gelman.diag(GR.BETA37)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.02

> gelman.diag(GR.BETA38)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA41)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.01

> gelman.diag(GR.BETA42)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA43)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA44)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.01

> gelman.diag(GR.BETA45)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1

> gelman.diag(GR.BETA46)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.02

> gelman.diag(GR.BETA47)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1       1.01

> gelman.diag(GR.BETA48)
Potential scale reduction
factors:

     Point est. Upper C.I.
[1,]          1          1
```

```
####################
### Prepare Data ###
####################

### Prepare Data ###

setwd("D:/DUKE/Courses/STA 290 Bayesian and Modern Statistics/Final Project")

raw <- read.fwf("usa_00006.dat", width = c(4, 2, 8, 10, 4, 1, 4, 10, 1, 3, 1, 1, 2, 1, 2,
7, 7, 4, 2, 3, 4, 4))
raw <- cbind(raw[ , 5], raw[ , 8:22])

colnames(raw) = c("LivingCity", "PersonalWeight", "Children<5", "Age", "Sex",
                  "Employment", "EmploymentD", "ClassWork", "ClassWorkD", "IncomeP",
                  "IncomeF", "WorkingCity", "Transportation", "Time", "Depart", "Arrive")

raw <- cbind(raw[ , 1:6], raw[ , 8], raw[ , 10:16])
attach(raw)

data <- subset(raw,
               ( LivingCity== 4610 | LivingCity== 1190 | LivingCity == 5330
                 LivingCity == 6290)
               & (WorkingCity == LivingCity) & Employment == 1 & ClassWork == 2
               & Time > 0 & Time < 120 & (Transportation == 31 | Transportation == 33
                                          |Transportation == 34 | Transportation == 35))
# the above: public transportation only
save(data, file = "data.Rdata")
detach(raw)

# Restrict to morning hours

data <- cbind(data[ , 2:5], data[ , 8:14])
data1 <- subset(data, data$Depart>659 & data$Depart<931 & data$Arrive > 729 &
                data$Arrive <1001)

# Randomly select
NYC <- subset(data1, data1$"WorkingCity" == 4610)
Chi <- subset(data1, data1$"WorkingCity" == 1190)
Phi <- subset(data1, data1$"WorkingCity" == 5330)
SF  <- subset(data1, data1$"WorkingCity" == 6290)

NYC <- NYC[sample(nrow(NYC),1000), ]
Chi <- Chi[sample(nrow(Chi),1000), ]
# Phi <- Phi[sample(nrow(Phi),300), ]
# SF  <- SF[sample(nrow(SF),300), ]

alldata <- rbind(NYC, Chi, Phi, SF)
save(alldata, file = "clear data.Rdata")

# Histograms

hist((alldata[ , 9]), breaks=20)
hist(log(alldata[ , 9]), breaks=20)

par(mfrow = c(2,2))
hist((NYC[ , 9]), breaks=15, main = "NYC")
hist((Chi[ , 9]), breaks=15, main = "Chicago")
hist((Phi[ , 9]), breaks=15, main = "Philadelphia")
hist((SF[ , 9]), breaks=15, main = "San Francisco")


hist(log(NYC[ , 9]), breaks=20, main = "NYC")
hist(log(Chi[ , 9]), breaks=20, main = "Chicago")
hist(log(Phi[ , 9]), breaks=20, main = "Philadelphia")
hist(log(SF[ , 9]), breaks=20, main = "San Francisco")
```

```
dev.off()

# Other EDAs

library(corpcor)
library(mvtnorm)
library(lattice)
library(coda)
library(MCMCpack)

par(mfrow = c(2,2))
boxplot(log(alldata$Time) ~ alldata$WorkingCity, ylab="logtime", xlab="1190=Chicago,
4610=NYC, 5330=Phi., 6290=SF")
boxplot(log(alldata$Time) ~ alldata$Sex, ylab="logtime", xlab="1=Male, 2=Female")
boxplot(log(alldata$Time) ~ alldata$Transportation, ylab = "logtime", xlab = "31=Bus,
33=Subway, 34=Railroad, 35=Taxi")
boxplot(log(alldata$Time) ~ alldata$Children, ylab="logtime", xlab="Number of Children
under 5 years old")

attach(alldata)
xyplot(log(Time)~log(IncomeP) | WorkingCity, main = "log(Time)~log(IncomeP) XYPlot by
WorkingCity")
xyplot(log(Time)~log(IncomeP) | Transportation, main = "log(Time)~log(IncomeP) XYPlot by
Transportation")
xyplot(log(IncomeP)~Transportation | WorkingCity, main = "log(IncomeP)~Transportation
XYPlot by WorkingCity")

# Some changes

Transportation=factor(Transportation)
WorkingCity=factor(WorkingCity)
Sex=factor(Sex)
levels(WorkingCity) = c("Chi", "NYC", "Phi","SF")
levels(Transportation)=c("Bus", "Subway", "Railroad", "Taxi")
levels(Sex)=c("Male", "Female")

##creating dummy variables and other preparations
SexD = model.matrix(~Sex-1)
alldata[11:12, "SexD"] <- NA
alldata$SexD <- SexD
TransportationD = model.matrix(~Transportation-1)
alldata[12:13, "TransportationD"] <- NA
alldata$TransportationD<-TransportationD

Tran1 <- TransportationD[, 2]
Tran1 <- TransportationD[, 1]
Tran2 <- TransportationD[, 2]
Tran3 <- TransportationD[, 3]
Tran4 <- TransportationD[, 4]
Sex <- SexD[ , 1]
data2 <- cbind(alldata[,2:3], Sex, alldata[,5:7], Tran1, Tran2, Tran3, Tran4, alldata[,
9:11])
alldata <- data2 [ , 1:11]
alldata <- cbind(alldata[ , 1:4], alldata[ , 6:11])
save(alldata, file = "clear data.Rdata")

###################
### MCMC Process ###
###################

### Prepare Prior ###

aa=log(alldata$IncomeP)
age=Age-mean(Age)
alldata=cbind(alldata, aa, age)
attach(alldata)
```

```
##find coeff for OLS

ols=lm(log(Time)~alldata$Children+age+Sex+log(IncomeP)+Tran1+Tran2+Tran4)

Y <- list()
X <- list()
N <- NULL
Y[[1]] <- alldata[alldata$WorkingCity == 1190, 10]
Y[[2]] <- alldata[alldata$WorkingCity == 4610, 10]
Y[[3]] <- alldata[alldata$WorkingCity == 5330, 10]
Y[[4]] <- alldata[alldata$WorkingCity == 6290, 10]
N[1] <- sum(alldata$WorkingCity == 1190)
N[2] <- sum(alldata$WorkingCity == 4610)
N[3] <- sum(alldata$WorkingCity == 5330)
N[4] <- sum(alldata$WorkingCity == 6290)
x11 <- alldata[alldata$WorkingCity == 1190, 1] #Children number
x21 <- alldata[alldata$WorkingCity == 1190, 12] #age
x31 <- alldata[alldata$WorkingCity == 1190, 3] #Sex
x41 <- alldata[alldata$WorkingCity == 1190, 11] #log Personal Income
x51 <- alldata[alldata$WorkingCity == 1190, 6] #Transportation--Bus
x61 <- alldata[alldata$WorkingCity == 1190, 7] #Transportation--Subway
x71 <- alldata[alldata$WorkingCity == 1190, 8] #Transportation--Railroad
x81 <- alldata[alldata$WorkingCity == 1190, 9] #Transportation--Taxi

x12 <- alldata[alldata$WorkingCity == 4610, 1] #Children number
x22 <- alldata[alldata$WorkingCity == 4610, 12] #age
x32 <- alldata[alldata$WorkingCity == 4610, 3] #Sex
x42 <- alldata[alldata$WorkingCity == 4610, 11] #log Personal Income
x52 <- alldata[alldata$WorkingCity == 4610, 6] #Transportation--Bus
x62 <- alldata[alldata$WorkingCity == 4610, 7] #Transportation--Subway
x72 <- alldata[alldata$WorkingCity == 4610, 8] #Transportation--Railroad
x82 <- alldata[alldata$WorkingCity == 4610, 9] #Transportation–Taxi

x13 <- alldata[alldata$WorkingCity == 5330, 1] #Children number
x23 <- alldata[alldata$WorkingCity == 5330, 12] #age
x33 <- alldata[alldata$WorkingCity == 5330, 3] #Sex
x43 <- alldata[alldata$WorkingCity == 5330, 11] #log Personal Income
x53 <- alldata[alldata$WorkingCity == 5330, 6] #Transportation--Bus
x63 <- alldata[alldata$WorkingCity == 5330, 7] #Transportation--Subway
x73 <- alldata[alldata$WorkingCity == 5330, 8] #Transportation--Railroad
x83 <- alldata[alldata$WorkingCity == 5330, 9] #Transportation–Taxi

x14 <- alldata[alldata$WorkingCity == 6290, 1] #Children number
x24 <- alldata[alldata$WorkingCity == 6290, 12] #age
x34 <- alldata[alldata$WorkingCity == 6290, 3] #Sex
x44 <- alldata[alldata$WorkingCity == 6290, 11] #log Personal Income
x54 <- alldata[alldata$WorkingCity == 6290, 6] #Transportation--Bus
x64 <- alldata[alldata$WorkingCity == 6290, 7] #Transportation--Subway
x74 <- alldata[alldata$WorkingCity == 6290, 8] #Transportation--Railroad
x84 <- alldata[alldata$WorkingCity == 6290, 9] #Transportation--Taxi

X[[1]] <- cbind(rep(1, N[1]),  x11, x21, x31, x41, x51, x61, x81)
X[[2]] <- cbind(rep(1, N[2]),  x12, x22, x32, x42, x52, x62, x82)
X[[3]] <- cbind(rep(1, N[3]),  x13, x23, x33, x43, x53, x63, x83)
X[[4]] <- cbind(rep(1, N[4]),  x14, x24, x34, x44, x54, x64, x84)

group1=subset(alldata, alldata$WorkingCity==1190)
group2=subset(alldata, alldata$WorkingCity==4610)
group3=subset(alldata, alldata$WorkingCity==5330)
group4=subset(alldata, alldata$WorkingCity==6290)

reg1 <- lm(log(Y[[1]])~-1+X[[1]])
reg2 <- lm(log(Y[[2]])~-1+X[[2]])
reg3 <- lm(log(Y[[3]])~-1+X[[3]])
reg4 <- lm(log(Y[[4]])~-1+X[[4]])
```

```
group1=subset(alldata, alldata$WorkingCity==1190)
group2=subset(alldata, alldata$WorkingCity==4610)
group3=subset(alldata, alldata$WorkingCity==5330)
group4=subset(alldata, alldata$WorkingCity==6290)


m=4

BETA.prior = matrix(NA, nrow=4, ncol=8)
BETA.prior[,1] = 4.2
BETA.prior[,2] = 0.06
BETA.prior[,3] = 0.004
BETA.prior[,4] = -0.012
BETA.prior[,5] = -0.069
BETA.prior[,6] = -0.21
BETA.prior[,7] = -0.07
BETA.prior[,8] = -1.0

mu0 = c(4.2, 0.06, 0.004, -0.012, -0.069, -0.21, -0.07, -1.0)

S0 = diag(8)
S0[3,3]=0.1

s2=1/(nrow(alldata)-
1)*(sum((reg1$resid)^2)+sum((reg2$resid)^2)+sum((reg3$resid)^2)+sum((reg4$resid)^2))
eta0 = 4
nu0 = 2
sigma20 = s2
iL0 = iSigma = solve(S0)


### Prepare MCMC ###

S=20000
n = c(nrow(group1),nrow(group2),nrow(group3),nrow(group4))
a=2
b = 2
THETA.post = NULL
SIGMA.post = array(NA, dim = c(8,8,S))
sigma20.post = matrix(NA, nrow=S+1, ncol=1)
sigma2.post = matrix(NA, nrow=S, ncol=m)
BETA.post = array(NA, dim = c(m,8,S+1))
X = matrix(NA, nrow=2923, ncol=8)
X[,1] = 1
X[,2] = alldata$Children
X[,3] = age
X[,4] = Sex
X[,5] = aa
X[,6] = Tran1
X[,7] = Tran2
X[,8] = Tran4

XX1 = t( X[1:n[1],]) %*% ( X[1:n[1],])
XY1 = t( X[1:n[1],]) %*% as.matrix(log(Time[1:n[1]]))
XX2 = t( X[(n[1]+1):(n[1]+n[2]),]) %*% ( X[(n[1]+1):(n[1]+n[2]),])
XY2 = t( X[(n[1]+1):(n[1]+n[2]),]) %*% as.matrix(log(Time[(n[1]+1):(n[1]+n[2])]))
XX3 = t( X[(n[1]+n[2]+1):(n[1]+n[2]+n[3]),]) %*% ( X[(n[1]+n[2]+1):(n[1]+n[2]+n[3]),])
XY3 = t(X[(n[1]+n[2]+1):(n[1]+n[2]+n[3]),])%*%as.matrix(log(Time[(n[1]+n[2]+1):
  (n[1] +n[2]+n[3])]))
XX4 = t( X[(n[1]+n[2]+n[3]+1):2923,]) %*% ( X[(n[1]+n[2]+n[3]+1):2923,])
XY4 = t(X[(n[1]+n[2]+n[3]+1):2923,])%*%as.matrix(log(Time[(n[1]+n[2]+n[3]+1):
  2923]))
SSR1 = sum((reg1$resid)^2)
SSR2 = sum((reg2$resid)^2)
SSR3 = sum((reg3$resid)^2)
SSR4 = sum((reg4$resid)^2)
```

```
sigma20.post[1] = s2
BETA.post[,,1] =BETA.prior

### start MCMC ###

for (s in 1:S){

  Lm = solve(iL0+m*iSigma)
  mum = Lm%*%(iL0%*%mu0 + iSigma%*%apply(BETA.post[,,s],2,sum))
  theta = t(rmvnorm(1,mum,Lm))

  mtheta = matrix(theta, m,8,byrow = TRUE)
  iSigma = rwish(8+m,
                 solve(S0+t(BETA.post[,,s]-mtheta)%*%(BETA.post[,,s]-mtheta))  )

  sigma2.post[s,1] = 1/rgamma(1, (nu0+n[1])/2, (nu0*sigma20.post[s]+SSR1)/2)
  sigma2.post[s,2] = 1/rgamma(1, (nu0+n[2])/2, (nu0*sigma20.post[s]+SSR2)/2)
  sigma2.post[s,3] = 1/rgamma(1, (nu0+n[3])/2, (nu0*sigma20.post[s]+SSR3)/2)
  sigma2.post[s,4] = 1/rgamma(1, (nu0+n[4])/2, (nu0*sigma20.post[s]+SSR4)/2)
  sigma20.post[s+1] = rgamma(1, (m*nu0/2+a), (nu0/2*sum(1/sigma2.post[s,])+b))

  beta.posterior = matrix(NA, nrow = m, ncol = 8)
  beta1.variance= solve(iSigma + XX1/sigma2.post[s,1])
  beta1.mean = beta1.variance %*% (iSigma %*% theta + XY1/sigma2.post[s,1])
  beta.posterior[1,] = rmvnorm(1,beta1.mean, beta1.variance)
  beta2.variance= solve(iSigma + XX2/sigma2.post[s,2])
  beta2.mean = beta2.variance %*% (iSigma %*% theta + XY2/sigma2.post[s,2])
  beta.posterior[2,] = rmvnorm(1,beta2.mean, beta2.variance)
  beta3.variance= solve(iSigma + XX3/sigma2.post[s,3])
  beta3.mean = beta3.variance %*% (iSigma %*% theta + XY3/sigma2.post[s,3])
  beta.posterior[3,] = rmvnorm(1,beta3.mean, beta3.variance)
  beta4.variance= solve(iSigma + XX4/sigma2.post[s,4])
  beta4.mean = beta4.variance %*% (iSigma %*% theta + XY4/sigma2.post[s,4])
  beta.posterior[4,] = rmvnorm(1,beta4.mean, beta4.variance)

  SSR1 = sum((log(Time[1:n[1]])-X[1:n[1],] * beta.posterior[1,])^2)
  SSR2 = sum((log(Time[(n[1]+1):(n[1]+n[2])])-X[(n[1]+1):(n[1]+n[2]),] *
    beta.posterior[2,])^2)
  SSR3 = sum((log(Time[(n[1]+n[2]+1):(n[1]+n[2]+n[3])])-X[(n[1]+n[2]+1):
    (n[1]+n[2]+n[3]),] * beta.posterior[3,])^2)
  SSR4 = sum((log(Time[(n[1]+n[2]+n[3]+1):2923])-X[(n[1]+n[2]+n[3]+1):2923,]*
    beta.posterior[4,])^2)

  THETA.post = rbind(THETA.post, t(theta))
  SIGMA.post [,,s]= solve(iSigma)
  BETA.post[,,s+1] = beta.posterior

}


########################
### Plots and Checks ###
########################

# Density of Posterior BETA for Group 1 - Chicago#

par(mfrow = c(3,3))

plot(density(BETA.post[1,1,]), xlab = expression(beta[11]), xlim = c(0, 10), ylim =
c(0,0.4), main =
  expression(paste("Density plot of ", beta[11])))
x1 = seq(-5, 15, length =1000)
lines(x=x1, y=dnorm(x1, BETA.prior[,1],sqrt(S0[1,1])), col ="gray")

plot(density(BETA.post[1,2,]), xlab = expression(beta[12]), xlim = c(-6, 6), ylim =
c(0,0.4), main =
```

```
  expression(paste("Density plot of ", beta[12])))
x2 = seq(-6, 6, length =1000)
lines(x=x2, y=dnorm(x2, BETA.prior[,2],sqrt(S0[2,2])), col ="gray")

plot(density(BETA.post[1,3,]), xlab = expression(beta[13]), xlim = c(-1, 1), main =
  expression(paste("Density plot of ", beta[13])))
x3 = seq(-1, 1, length =1000)
lines(x=x3, y=dnorm(x3, BETA.prior[,3],sqrt(S0[3,3])), col ="gray")

plot(density(BETA.post[1,4,]), xlab = expression(beta[14]), xlim = c(-6, 6), ylim =
c(0,0.4),main =
  expression(paste("Density plot of ", beta[14])))
x4 = seq(-6, 6, length =1000)
lines(x=x4, y=dnorm(x4, BETA.prior[,4],sqrt(S0[4,4])), col ="gray")

plot(density(BETA.post[1,5,]), xlab = expression(beta[15]), xlim = c(-3, 3), main =
  expression(paste("Density plot of ", beta[15])))
x5 = seq(-3, 3, length =1000)
lines(x=x5, y=dnorm(x5, BETA.prior[,5],sqrt(S0[5,5])), col ="gray")

plot(density(BETA.post[1,6,]), xlab = expression(beta[16]), xlim = c(-6, 6), ylim = c(0,
0.4), main =
  expression(paste("Density plot of ", beta[16])))
x6 = seq(-6, 6, length =1000)
lines(x=x6, y=dnorm(x6, BETA.prior[,6],sqrt(S0[6,6])), col ="gray")

plot(density(BETA.post[1,7,]), xlab = expression(beta[17]), xlim = c(-5, 5), ylim = c(0,
0.4), main =
  expression(paste("Density plot of ", beta[17])))
x7 = seq(-5, 5, length =1000)
lines(x=x7, y=dnorm(x7, BETA.prior[,7],sqrt(S0[7,7])), col ="gray")

plot(density(BETA.post[1,8,]), xlab = expression(beta[18]), xlim = c(-7, 5), ylim = c(0,
0.4), main =
  expression(paste("Density plot of ", beta[18])))
x8 = seq(-5, 5, length =1000)
lines(x=x8, y=dnorm(x8, BETA.prior[,8],sqrt(S0[8,8])), col ="gray")

# Density of Posterior BETA for Group 2 - NYC #

par(mfrow = c(3,3))

plot(density(BETA.post[2,1,]), xlab = expression(beta[21]), xlim = c(0,10), ylim = c(0,
0.4), main =
  expression(paste("Density plot of ", beta[21])))
x1 = seq(-5, 15, length =1000)
lines(x=x1, y=dnorm(x1, BETA.prior[,1],sqrt(S0[1,1])), col ="gray")

plot(density(BETA.post[2,2,]), xlab = expression(beta[22]), main =
  expression(paste("Density plot of ", beta[22])))
x2 = seq(-6, 6, length =1000)
lines(x=x2, y=dnorm(x2, BETA.prior[,2],sqrt(S0[2,2])), col ="gray")

plot(density(BETA.post[2,3,]), xlab = expression(beta[23]), main =
  expression(paste("Density plot of ", beta[23])))
x3 = seq(-1, 1, length =1000)
lines(x=x3, y=dnorm(x3, BETA.prior[,3],sqrt(S0[3,3])), col ="gray")

plot(density(BETA.post[2,4,]), xlab = expression(beta[24]), main =
  expression(paste("Density plot of ", beta[24])))
x4 = seq(-6, 6, length =1000)
lines(x=x4, y=dnorm(x4, BETA.prior[,4],sqrt(S0[4,4])), col ="gray")

plot(density(BETA.post[2,5,]), xlab = expression(beta[25]), main =
  expression(paste("Density plot of ", beta[25])))
x5 = seq(-3, 3, length =1000)
```

```
lines(x=x5, y=dnorm(x5, BETA.prior[,5],sqrt(S0[5,5])), col ="gray")

plot(density(BETA.post[2,6,]), xlab = expression(beta[26]), main =
  expression(paste("Density plot of ", beta[26])))
x6 = seq(-6, 6, length =1000)
lines(x=x6, y=dnorm(x6, BETA.prior[,6],sqrt(S0[6,6])), col ="gray")

plot(density(BETA.post[2,7,]), xlab = expression(beta[27]), main =
  expression(paste("Density plot of ", beta[27])))
x7 = seq(-5, 5, length =1000)
lines(x=x7, y=dnorm(x7, BETA.prior[,7],sqrt(S0[7,7])), col ="gray")

plot(density(BETA.post[2,8,]), xlab = expression(beta[28]), xlim = c(-7, 5), ylim = c(0,
0.4), main =
  expression(paste("Density plot of ", beta[28])))
x8 = seq(-5, 5, length =1000)
lines(x=x8, y=dnorm(x8, BETA.prior[,8],sqrt(S0[8,8])), col ="gray")

# Density of Posterior BETA for Group 3 - Philadelphia #

par(mfrow = c(3,3))

plot(density(BETA.post[3,1,]), xlab = expression(beta[31]), xlim = c(0,10), ylim = c(0,
0.4), main =
  expression(paste("Density plot of ", beta[31])))
x1 = seq(-5, 15, length =1000)
lines(x=x1, y=dnorm(x1, BETA.prior[,1],sqrt(S0[1,1])), col ="gray")

plot(density(BETA.post[3,2,]), xlab = expression(beta[32]), main =
  expression(paste("Density plot of ", beta[32])))
x2 = seq(-6, 6, length =1000)
lines(x=x2, y=dnorm(x2, BETA.prior[,2],sqrt(S0[2,2])), col ="gray")

plot(density(BETA.post[3,3,]), xlab = expression(beta[33]), main =
  expression(paste("Density plot of ", beta[33])))
x3 = seq(-1, 1, length =1000)
lines(x=x3, y=dnorm(x3, BETA.prior[,3],sqrt(S0[3,3])), col ="gray")

plot(density(BETA.post[3,4,]), xlab = expression(beta[34]), main =
  expression(paste("Density plot of ", beta[34])))
x4 = seq(-6, 6, length =1000)
lines(x=x4, y=dnorm(x4, BETA.prior[,4],sqrt(S0[4,4])), col ="gray")

plot(density(BETA.post[3,5,]), xlab = expression(beta[35]), main =
  expression(paste("Density plot of ", beta[35])))
x5 = seq(-3, 3, length =1000)
lines(x=x5, y=dnorm(x5, BETA.prior[,5],sqrt(S0[5,5])), col ="gray")

plot(density(BETA.post[3,6,]), xlab = expression(beta[36]), main =
  expression(paste("Density plot of ", beta[36])))
x6 = seq(-6, 6, length =1000)
lines(x=x6, y=dnorm(x6, BETA.prior[,6],sqrt(S0[6,6])), col ="gray")

plot(density(BETA.post[3,7,]), xlab = expression(beta[37]), main =
  expression(paste("Density plot of ", beta[37])))
x7 = seq(-5, 5, length =1000)
lines(x=x7, y=dnorm(x7, BETA.prior[,7],sqrt(S0[7,7])), col ="gray")

plot(density(BETA.post[3,8,]), xlab = expression(beta[38]), xlim = c(-7, 5), ylim = c(0,
0.4), main =
  expression(paste("Density plot of ", beta[38])))
x8 = seq(-5, 5, length =1000)
lines(x=x8, y=dnorm(x8, BETA.prior[,8],sqrt(S0[8,8])), col ="gray")

# Density of Posterior BETA for Group 4 - San Francisco #
```

```
par(mfrow = c(3,3))

plot(density(BETA.post[4,1,]), xlab = expression(beta[41]), xlim = c(0,10), ylim = c(0,
0.4), main =
  expression(paste("Density plot of ", beta[41])))
x1 = seq(-5, 15, length =1000)
lines(x=x1, y=dnorm(x1, BETA.prior[,1],sqrt(S0[1,1])), col ="gray")

plot(density(BETA.post[4,2,]), xlab = expression(beta[42]), main =
  expression(paste("Density plot of ", beta[42])))
x2 = seq(-6, 6, length =1000)
lines(x=x2, y=dnorm(x2, BETA.prior[,2],sqrt(S0[2,2])), col ="gray")

plot(density(BETA.post[4,3,]), xlab = expression(beta[43]), main =
  expression(paste("Density plot of ", beta[43])))
x3 = seq(-1, 1, length =1000)
lines(x=x3, y=dnorm(x3, BETA.prior[,3],sqrt(S0[3,3])), col ="gray")

plot(density(BETA.post[4,4,]), xlab = expression(beta[44]), main =
  expression(paste("Density plot of ", beta[44])))
x4 = seq(-6, 6, length =1000)
lines(x=x4, y=dnorm(x4, BETA.prior[,4],sqrt(S0[4,4])), col ="gray")

plot(density(BETA.post[4,5,]), xlab = expression(beta[45]), main =
  expression(paste("Density plot of ", beta[45])))
x5 = seq(-3, 3, length =1000)
lines(x=x5, y=dnorm(x5, BETA.prior[,5],sqrt(S0[5,5])), col ="gray")

plot(density(BETA.post[4,6,]), xlab = expression(beta[46]), main =
  expression(paste("Density plot of ", beta[46])))
x6 = seq(-6, 6, length =1000)
lines(x=x6, y=dnorm(x6, BETA.prior[,6],sqrt(S0[6,6])), col ="gray")

plot(density(BETA.post[4,7,]), xlab = expression(beta[47]), main =
  expression(paste("Density plot of ", beta[47])))
x7 = seq(-5, 5, length =1000)
lines(x=x7, y=dnorm(x7, BETA.prior[,7],sqrt(S0[7,7])), col ="gray")

plot(density(BETA.post[4,8,]), xlab = expression(beta[48]), xlim = c(-7, 5), ylim = c(0,
0.4), main =
  expression(paste("Density plot of ", beta[48])))
x8 = seq(-5, 5, length =1000)
lines(x=x8, y=dnorm(x8, BETA.prior[,8],sqrt(S0[8,8])), col ="gray")

### Check Convergence ###

# Group 1 - Chi. #

par(mfrow = c(3,3))
traceplot(as.mcmc(BETA.post[1,1,10001:20000]),
          ylab = expression(beta[11]))
traceplot(as.mcmc(BETA.post[1,2,10001:20000]),
          ylab = expression(beta[12]))
traceplot(as.mcmc(BETA.post[1,3,10001:20000]),
          ylab = expression(beta[13]))
traceplot(as.mcmc(BETA.post[1,4,10001:20000]),
          ylab = expression(beta[14]))
traceplot(as.mcmc(BETA.post[1,5,10001:20000]),
          ylab = expression(beta[15]))
traceplot(as.mcmc(BETA.post[1,6,10001:20000]),
          ylab = expression(beta[16]))
traceplot(as.mcmc(BETA.post[1,7,10001:20000]),
          ylab = expression(beta[17]))
traceplot(as.mcmc(BETA.post[1,8,10001:20000]),
          ylab = expression(beta[18]))
```

```
# Group 2 - NYC #

par(mfrow = c(3,3))
traceplot(as.mcmc(BETA.post[2,1,10001:20000]),
          ylab = expression(beta[21]))
traceplot(as.mcmc(BETA.post[2,2,10001:20000]),
          ylab = expression(beta[22]))
traceplot(as.mcmc(BETA.post[2,3,10001:20000]),
          ylab = expression(beta[23]))
traceplot(as.mcmc(BETA.post[2,4,10001:20000]),
          ylab = expression(beta[24]))
traceplot(as.mcmc(BETA.post[2,5,10001:20000]),
          ylab = expression(beta[25]))
traceplot(as.mcmc(BETA.post[2,6,10001:20000]),
          ylab = expression(beta[26]))
traceplot(as.mcmc(BETA.post[2,7,10001:20000]),
          ylab = expression(beta[27]))
traceplot(as.mcmc(BETA.post[2,8,10001:20000]),
          ylab = expression(beta[28]))

# Group 3 - Phi. #

par(mfrow = c(3,3))
traceplot(as.mcmc(BETA.post[3,1,10001:20000]),
          ylab = expression(beta[31]))
traceplot(as.mcmc(BETA.post[3,2,10001:20000]),
          ylab = expression(beta[32]))
traceplot(as.mcmc(BETA.post[3,3,10001:20000]),
          ylab = expression(beta[33]))
traceplot(as.mcmc(BETA.post[3,4,10001:20000]),
          ylab = expression(beta[34]))
traceplot(as.mcmc(BETA.post[3,5,10001:20000]),
          ylab = expression(beta[35]))
traceplot(as.mcmc(BETA.post[3,6,10001:20000]),
          ylab = expression(beta[36]))
traceplot(as.mcmc(BETA.post[3,7,10001:20000]),
          ylab = expression(beta[37]))
traceplot(as.mcmc(BETA.post[3,8,10001:20000]),
          ylab = expression(beta[38]))

# Group 4 - SF. #

par(mfrow = c(3,3))
traceplot(as.mcmc(BETA.post[4,1,10001:20000]),
          ylab = expression(beta[41]))
traceplot(as.mcmc(BETA.post[4,2,10001:20000]),
          ylab = expression(beta[42]))
traceplot(as.mcmc(BETA.post[4,3,10001:20000]),
          ylab = expression(beta[43]))
traceplot(as.mcmc(BETA.post[4,4,10001:20000]),
          ylab = expression(beta[44]))
traceplot(as.mcmc(BETA.post[4,5,10001:20000]),
          ylab = expression(beta[45]))
traceplot(as.mcmc(BETA.post[4,6,10001:20000]),
          ylab = expression(beta[46]))
traceplot(as.mcmc(BETA.post[4,7,10001:20000]),
          ylab = expression(beta[47]))
traceplot(as.mcmc(BETA.post[4,8,10001:20000]),
          ylab = expression(beta[48]))

### Check AutoCorrelation ###

autocorr(as.mcmc(BETA.post[1,1,10000:20000]))
autocorr(as.mcmc(BETA.post[1,2,10000:20000]))
autocorr(as.mcmc(BETA.post[1,3,10000:20000]))
autocorr(as.mcmc(BETA.post[1,4,10000:20000]))
```

```
autocorr(as.mcmc(BETA.post[1,5,10000:20000]))
autocorr(as.mcmc(BETA.post[1,6,10000:20000]))
autocorr(as.mcmc(BETA.post[1,7,10000:20000]))
autocorr(as.mcmc(BETA.post[1,8,10000:20000]))

autocorr(as.mcmc(BETA.post[2,1,10000:20000]))
autocorr(as.mcmc(BETA.post[2,2,10000:20000]))
autocorr(as.mcmc(BETA.post[2,3,10000:20000]))
autocorr(as.mcmc(BETA.post[2,4,10000:20000]))
autocorr(as.mcmc(BETA.post[2,5,10000:20000]))
autocorr(as.mcmc(BETA.post[2,6,10000:20000]))
autocorr(as.mcmc(BETA.post[2,7,10000:20000]))
autocorr(as.mcmc(BETA.post[2,8,10000:20000]))

autocorr(as.mcmc(BETA.post[3,1,10000:20000]))
autocorr(as.mcmc(BETA.post[3,2,10000:20000]))
autocorr(as.mcmc(BETA.post[3,3,10000:20000]))
autocorr(as.mcmc(BETA.post[3,4,10000:20000]))
autocorr(as.mcmc(BETA.post[3,5,10000:20000]))
autocorr(as.mcmc(BETA.post[3,6,10000:20000]))
autocorr(as.mcmc(BETA.post[3,7,10000:20000]))
autocorr(as.mcmc(BETA.post[3,8,10000:20000]))

autocorr(as.mcmc(BETA.post[4,1,10000:20000]))
autocorr(as.mcmc(BETA.post[4,2,10000:20000]))
autocorr(as.mcmc(BETA.post[4,3,10000:20000]))
autocorr(as.mcmc(BETA.post[4,4,10000:20000]))
autocorr(as.mcmc(BETA.post[4,5,10000:20000]))
autocorr(as.mcmc(BETA.post[4,6,10000:20000]))
autocorr(as.mcmc(BETA.post[4,7,10000:20000]))
autocorr(as.mcmc(BETA.post[4,8,10000:20000]))

###plot acf

par(mfrow = c(3,3))
acf(as.mcmc(BETA.post[1,1,10000:20000]), main = expression(beta[11]))
acf(as.mcmc(BETA.post[1,2,10000:20000]), main = expression(beta[12]))
acf(as.mcmc(BETA.post[1,3,10000:20000]), main = expression(beta[13]))
acf(as.mcmc(BETA.post[1,4,10000:20000]), main = expression(beta[14]))
acf(as.mcmc(BETA.post[1,5,10000:20000]), main = expression(beta[15]))
acf(as.mcmc(BETA.post[1,6,10000:20000]), main = expression(beta[16]))
acf(as.mcmc(BETA.post[1,7,10000:20000]), main = expression(beta[17]))
acf(as.mcmc(BETA.post[1,8,10000:20000]), main = expression(beta[18]))

par(mfrow = c(3,3))
acf(as.mcmc(BETA.post[2,1,10000:20000]), main = expression(beta[21]))
acf(as.mcmc(BETA.post[2,2,10000:20000]), main = expression(beta[22]))
acf(as.mcmc(BETA.post[2,3,10000:20000]), main = expression(beta[23]))
acf(as.mcmc(BETA.post[2,4,10000:20000]), main = expression(beta[24]))
acf(as.mcmc(BETA.post[2,5,10000:20000]), main = expression(beta[25]))
acf(as.mcmc(BETA.post[2,6,10000:20000]), main = expression(beta[26]))
acf(as.mcmc(BETA.post[2,7,10000:20000]), main = expression(beta[27]))
acf(as.mcmc(BETA.post[2,8,10000:20000]), main = expression(beta[28]))

par(mfrow = c(3,3))
acf(as.mcmc(BETA.post[3,1,10000:20000]), main = expression(beta[31]))
acf(as.mcmc(BETA.post[3,2,10000:20000]), main = expression(beta[32]))
acf(as.mcmc(BETA.post[3,3,10000:20000]), main = expression(beta[33]))
acf(as.mcmc(BETA.post[3,4,10000:20000]), main = expression(beta[34]))
acf(as.mcmc(BETA.post[3,5,10000:20000]), main = expression(beta[35]))
acf(as.mcmc(BETA.post[3,6,10000:20000]), main = expression(beta[36]))
acf(as.mcmc(BETA.post[3,7,10000:20000]), main = expression(beta[37]))
acf(as.mcmc(BETA.post[3,8,10000:20000]), main = expression(beta[38]))

par(mfrow = c(3,3))
acf(as.mcmc(BETA.post[4,1,10000:20000]), main = expression(beta[41]))
```

```
acf(as.mcmc(BETA.post[4,2,10000:20000]), main = expression(beta[42]))
acf(as.mcmc(BETA.post[4,3,10000:20000]), main = expression(beta[43]))
acf(as.mcmc(BETA.post[4,4,10000:20000]), main = expression(beta[44]))
acf(as.mcmc(BETA.post[4,5,10000:20000]), main = expression(beta[45]))
acf(as.mcmc(BETA.post[4,6,10000:20000]), main = expression(beta[46]))
acf(as.mcmc(BETA.post[4,7,10000:20000]), main = expression(beta[47]))
acf(as.mcmc(BETA.post[4,8,10000:20000]), main = expression(beta[48]))


### Gelman-Rubin Diagnostic ###
### Start from different value
###BETA.post2 = array(NA, dim = c(m,8,S+1))
###BETA.post2[,,1] = 1

Y <- list()
X <- list()
N <- NULL
Y[[1]] <- alldata[alldata$WorkingCity == 1190, 10]
Y[[2]] <- alldata[alldata$WorkingCity == 4610, 10]
Y[[3]] <- alldata[alldata$WorkingCity == 5330, 10]
Y[[4]] <- alldata[alldata$WorkingCity == 6290, 10]
N[1] <- sum(alldata$WorkingCity == 1190)
N[2] <- sum(alldata$WorkingCity == 4610)
N[3] <- sum(alldata$WorkingCity == 5330)
N[4] <- sum(alldata$WorkingCity == 6290)
x11 <- alldata[alldata$WorkingCity == 1190, 1] #Children number
x21 <- alldata[alldata$WorkingCity == 1190, 12] #age
x31 <- alldata[alldata$WorkingCity == 1190, 3] #Sex
x41 <- alldata[alldata$WorkingCity == 1190, 11] #log Personal Income
x51 <- alldata[alldata$WorkingCity == 1190, 6] #Transportation--Bus
x61 <- alldata[alldata$WorkingCity == 1190, 7] #Transportation--Subway
x71 <- alldata[alldata$WorkingCity == 1190, 8] #Transportation--Railroad
x81 <- alldata[alldata$WorkingCity == 1190, 9] #Transportation--Taxi

x12 <- alldata[alldata$WorkingCity == 4610, 1] #Children number
x22 <- alldata[alldata$WorkingCity == 4610, 12] #age
x32 <- alldata[alldata$WorkingCity == 4610, 3] #Sex
x42 <- alldata[alldata$WorkingCity == 4610, 11] #log Personal Income
x52 <- alldata[alldata$WorkingCity == 4610, 6] #Transportation--Bus
x62 <- alldata[alldata$WorkingCity == 4610, 7] #Transportation--Subway
x72 <- alldata[alldata$WorkingCity == 4610, 8] #Transportation--Railroad
x82 <- alldata[alldata$WorkingCity == 4610, 9] #Transportation—Taxi

x13 <- alldata[alldata$WorkingCity == 5330, 1] #Children number
x23 <- alldata[alldata$WorkingCity == 5330, 12] #age
x33 <- alldata[alldata$WorkingCity == 5330, 3] #Sex
x43 <- alldata[alldata$WorkingCity == 5330, 11] #log Personal Income
x53 <- alldata[alldata$WorkingCity == 5330, 6] #Transportation--Bus
x63 <- alldata[alldata$WorkingCity == 5330, 7] #Transportation--Subway
x73 <- alldata[alldata$WorkingCity == 5330, 8] #Transportation--Railroad
x83 <- alldata[alldata$WorkingCity == 5330, 9] #Transportation—Taxi

x14 <- alldata[alldata$WorkingCity == 6290, 1] #Children number
x24 <- alldata[alldata$WorkingCity == 6290, 12] #age
x34 <- alldata[alldata$WorkingCity == 6290, 3] #Sex
x44 <- alldata[alldata$WorkingCity == 6290, 11] #log Personal Income
x54 <- alldata[alldata$WorkingCity == 6290, 6] #Transportation--Bus
x64 <- alldata[alldata$WorkingCity == 6290, 7] #Transportation--Subway
x74 <- alldata[alldata$WorkingCity == 6290, 8] #Transportation--Railroad
x84 <- alldata[alldata$WorkingCity == 6290, 9] #Transportation--Taxi

X[[1]] <- cbind(rep(1, N[1]),  x11, x21, x31, x41, x51, x61, x81)
X[[2]] <- cbind(rep(1, N[2]),  x12, x22, x32, x42, x52, x62, x82)
X[[3]] <- cbind(rep(1, N[3]),  x13, x23, x33, x43, x53, x63, x83)
X[[4]] <- cbind(rep(1, N[4]),  x14, x24, x34, x44, x54, x64, x84)
```

```
group1=subset(alldata, alldata$WorkingCity==1190)
group2=subset(alldata, alldata$WorkingCity==4610)
group3=subset(alldata, alldata$WorkingCity==5330)
group4=subset(alldata, alldata$WorkingCity==6290)

reg1 <- lm(log(Y[[1]])~-1+X[[1]])
reg2 <- lm(log(Y[[2]])~-1+X[[2]])
reg3 <- lm(log(Y[[3]])~-1+X[[3]])
reg4 <- lm(log(Y[[4]])~-1+X[[4]])
group1=subset(alldata, alldata$WorkingCity==1190)
group2=subset(alldata, alldata$WorkingCity==4610)
group3=subset(alldata, alldata$WorkingCity==5330)
group4=subset(alldata, alldata$WorkingCity==6290)
m=4

BETA.prior = matrix(NA, nrow=4, ncol=8)
BETA.prior[,1] = 4.2
BETA.prior[,2] = 0.06
BETA.prior[,3] = 0.004
BETA.prior[,4] = -0.012
BETA.prior[,5] = -0.069
BETA.prior[,6] = -0.21
BETA.prior[,7] = -0.07
BETA.prior[,8] = -1.0

mu0 = c(4.2, 0.06, 0.004, -0.012, -0.069, -0.21, -0.07, -1.0)

S0 = diag(8)
S0[3,3]=0.1

s2=1/(nrow(alldata)-
1)*(sum((reg1$resid)^2)+sum((reg2$resid)^2)+sum((reg3$resid)^2)+sum((reg4$resid)^2))
eta0 = 4
nu0 = 2
sigma20 = s2
iL0 = iSigma = solve(S0)

S=20000
n = c(nrow(group1),nrow(group2),nrow(group3),nrow(group4))
a=2
b = 2
THETA.post = NULL
SIGMA.post = array(NA, dim = c(8,8,S))
sigma20.post = matrix(NA, nrow=S+1, ncol=1)
sigma2.post = matrix(NA, nrow=S, ncol=m)
BETA.post2 = array(NA, dim = c(m,8,S+1))
BETA.post2[,,1] = 1
X = matrix(NA, nrow=2923, ncol=8)
X[,1] = 1
X[,2] = alldata$Children
X[,3] = age
X[,4] = Sex
X[,5] = aa
X[,6] = Tran1
X[,7] = Tran2
X[,8] = Tran4

XX1 = t( X[1:n[1],]) %*% ( X[1:n[1],])
XY1 = t( X[1:n[1],]) %*% as.matrix(log(Time[1:n[1]]))
XX2 = t( X[(n[1]+1):(n[1]+n[2]),]) %*% ( X[(n[1]+1):(n[1]+n[2]),])
XY2 = t( X[(n[1]+1):(n[1]+n[2]),]) %*% as.matrix(log(Time[(n[1]+1):(n[1]+n[2])]))
XX3 = t( X[(n[1]+n[2]+1):(n[1]+n[2]+n[3]),]) %*% ( X[(n[1]+n[2]+1):(n[1]+n[2]+n[3]),])
XY3 = t(X[(n[1]+n[2]+1):(n[1]+n[2]+n[3]),])%*%as.matrix(log(Time[(n[1]+n[2]+1):
(n[1] +n[2]+n[3])]))
XX4 = t( X[(n[1]+n[2]+n[3]+1):2923,]) %*% ( X[(n[1]+n[2]+n[3]+1):2923,])
XY4 = t(X[(n[1]+n[2]+n[3]+1):2923,])%*%as.matrix(log(Time[(n[1]+n[2]+n[3]+1):
```

```
2923]))
SSR1 = sum((reg1$resid)^2)
SSR2 = sum((reg2$resid)^2)
SSR3 = sum((reg3$resid)^2)
SSR4 = sum((reg4$resid)^2)
sigma20.post[1] = s2
BETA.post2[,,1] =BETA.prior

for (s in 1:S){

      Lm = solve(iL0+m*iSigma)
      mum = Lm%*%(iL0%*%mu0 + iSigma%*%apply(BETA.post2[,,s],2,sum))
      theta = t(rmvnorm(1,mum,Lm))

      mtheta = matrix(theta, m,8,byrow = TRUE)
      iSigma = rwish(8+m,
                           solve(S0+t(BETA.post2[,,s]-mtheta)%*%(BETA.post2[,,s]-
mtheta))  )

      sigma2.post[s,1] = 1/rgamma(1, (nu0+n[1])/2, (nu0*sigma20.post[s]+SSR1)/2)
      sigma2.post[s,2] = 1/rgamma(1, (nu0+n[2])/2, (nu0*sigma20.post[s]+SSR2)/2)
      sigma2.post[s,3] = 1/rgamma(1, (nu0+n[3])/2, (nu0*sigma20.post[s]+SSR3)/2)
      sigma2.post[s,4] = 1/rgamma(1, (nu0+n[4])/2, (nu0*sigma20.post[s]+SSR4)/2)
      sigma20.post[s+1] = rgamma(1, (m*nu0/2+a), (nu0/2*sum(1/sigma2.post[s,])+b))

      beta.posterior = matrix(NA, nrow = m, ncol = 8)
      beta1.variance= solve(iSigma + XX1/sigma2.post[s,1])
      beta1.mean = beta1.variance %*% (iSigma %*% theta + XY1/sigma2.post[s,1])
      beta.posterior[1,] = rmvnorm(1,beta1.mean, beta1.variance)
      beta2.variance= solve(iSigma + XX2/sigma2.post[s,2])
      beta2.mean = beta2.variance %*% (iSigma %*% theta + XY2/sigma2.post[s,2])
      beta.posterior[2,] = rmvnorm(1,beta2.mean, beta2.variance)
      beta3.variance= solve(iSigma + XX3/sigma2.post[s,3])
      beta3.mean = beta3.variance %*% (iSigma %*% theta + XY3/sigma2.post[s,3])
      beta.posterior[3,] = rmvnorm(1,beta3.mean, beta3.variance)
      beta4.variance= solve(iSigma + XX4/sigma2.post[s,4])
      beta4.mean = beta4.variance %*% (iSigma %*% theta + XY4/sigma2.post[s,4])
      beta.posterior[4,] = rmvnorm(1,beta4.mean, beta4.variance)

      SSR1 = sum((log(Time[1:n[1]])-X[1:n[1],] * beta.posterior[1,])^2)
      SSR2 = sum((log(Time[(n[1]+1):(n[1]+n[2])])-X[(n[1]+1):(n[1]+n[2]),] *
                        beta.posterior[2,])^2)
      SSR3 = sum((log(Time[(n[1]+n[2]+1):(n[1]+n[2]+n[3])])-X[(n[1]+n[2]+1):
                        (n[1]+n[2]+n[3]),] * beta.posterior[3,])^2)
      SSR4 = sum((log(Time[(n[1]+n[2]+n[3]+1):2923])-X[(n[1]+n[2]+n[3]+1):2923,]*
                        beta.posterior[4,])^2)

      THETA.post = rbind(THETA.post, t(theta))
  SIGMA.post [,,s]= solve(iSigma)
  BETA.post2[,,s+1] = beta.posterior

}


GR.BETA11 = mcmc.list(as.mcmc(BETA.post[1,1,]), as.mcmc(BETA.post2[1,1,]))
GR.BETA12 = mcmc.list(as.mcmc(BETA.post[1,2,]), as.mcmc(BETA.post2[1,2,]))
GR.BETA13 = mcmc.list(as.mcmc(BETA.post[1,3,]), as.mcmc(BETA.post2[1,3,]))
GR.BETA14 = mcmc.list(as.mcmc(BETA.post[1,4,]), as.mcmc(BETA.post2[1,4,]))
GR.BETA15 = mcmc.list(as.mcmc(BETA.post[1,5,]), as.mcmc(BETA.post2[1,5,]))
GR.BETA16 = mcmc.list(as.mcmc(BETA.post[1,6,]), as.mcmc(BETA.post2[1,6,]))
GR.BETA17 = mcmc.list(as.mcmc(BETA.post[1,7,]), as.mcmc(BETA.post2[1,7,]))
GR.BETA18 = mcmc.list(as.mcmc(BETA.post[1,8,]), as.mcmc(BETA.post2[1,8,]))

gelman.diag(GR.BETA11)
gelman.diag(GR.BETA12)
gelman.diag(GR.BETA13)
```

```
gelman.diag(GR.BETA14)
gelman.diag(GR.BETA15)
gelman.diag(GR.BETA16)
gelman.diag(GR.BETA17)
gelman.diag(GR.BETA18)

GR.BETA21 = mcmc.list(as.mcmc(BETA.post[2,1,]), as.mcmc(BETA.post2[2,1,]))
GR.BETA22 = mcmc.list(as.mcmc(BETA.post[2,2,]), as.mcmc(BETA.post2[2,2,]))
GR.BETA23 = mcmc.list(as.mcmc(BETA.post[2,3,]), as.mcmc(BETA.post2[2,3,]))
GR.BETA24 = mcmc.list(as.mcmc(BETA.post[2,4,]), as.mcmc(BETA.post2[2,4,]))
GR.BETA25 = mcmc.list(as.mcmc(BETA.post[2,5,]), as.mcmc(BETA.post2[2,5,]))
GR.BETA26 = mcmc.list(as.mcmc(BETA.post[2,6,]), as.mcmc(BETA.post2[2,6,]))
GR.BETA27 = mcmc.list(as.mcmc(BETA.post[2,7,]), as.mcmc(BETA.post2[2,7,]))
GR.BETA28 = mcmc.list(as.mcmc(BETA.post[2,8,]), as.mcmc(BETA.post2[2,8,]))

gelman.diag(GR.BETA21)
gelman.diag(GR.BETA22)
gelman.diag(GR.BETA23)
gelman.diag(GR.BETA24)
gelman.diag(GR.BETA25)
gelman.diag(GR.BETA26)
gelman.diag(GR.BETA27)
gelman.diag(GR.BETA28)

GR.BETA31 = mcmc.list(as.mcmc(BETA.post[3,1,]), as.mcmc(BETA.post2[3,1,]))
GR.BETA32 = mcmc.list(as.mcmc(BETA.post[3,2,]), as.mcmc(BETA.post2[3,2,]))
GR.BETA33 = mcmc.list(as.mcmc(BETA.post[3,3,]), as.mcmc(BETA.post2[3,3,]))
GR.BETA34 = mcmc.list(as.mcmc(BETA.post[3,4,]), as.mcmc(BETA.post2[3,4,]))
GR.BETA35 = mcmc.list(as.mcmc(BETA.post[3,5,]), as.mcmc(BETA.post2[3,5,]))
GR.BETA36 = mcmc.list(as.mcmc(BETA.post[3,6,]), as.mcmc(BETA.post2[3,6,]))
GR.BETA37 = mcmc.list(as.mcmc(BETA.post[3,7,]), as.mcmc(BETA.post2[3,7,]))
GR.BETA38 = mcmc.list(as.mcmc(BETA.post[3,8,]), as.mcmc(BETA.post2[3,8,]))

gelman.diag(GR.BETA31)
gelman.diag(GR.BETA32)
gelman.diag(GR.BETA33)
gelman.diag(GR.BETA34)
gelman.diag(GR.BETA35)
gelman.diag(GR.BETA36)
gelman.diag(GR.BETA37)
gelman.diag(GR.BETA38)

GR.BETA41 = mcmc.list(as.mcmc(BETA.post[4,1,]), as.mcmc(BETA.post2[4,1,]))
GR.BETA42 = mcmc.list(as.mcmc(BETA.post[4,2,]), as.mcmc(BETA.post2[4,2,]))
GR.BETA43 = mcmc.list(as.mcmc(BETA.post[4,3,]), as.mcmc(BETA.post2[4,3,]))
GR.BETA44 = mcmc.list(as.mcmc(BETA.post[4,4,]), as.mcmc(BETA.post2[4,4,]))
GR.BETA45 = mcmc.list(as.mcmc(BETA.post[4,5,]), as.mcmc(BETA.post2[4,5,]))
GR.BETA46 = mcmc.list(as.mcmc(BETA.post[4,6,]), as.mcmc(BETA.post2[4,6,]))
GR.BETA47 = mcmc.list(as.mcmc(BETA.post[4,7,]), as.mcmc(BETA.post2[4,7,]))
GR.BETA48 = mcmc.list(as.mcmc(BETA.post[4,8,]), as.mcmc(BETA.post2[4,8,]))

gelman.diag(GR.BETA41)
gelman.diag(GR.BETA42)
gelman.diag(GR.BETA43)
gelman.diag(GR.BETA44)
gelman.diag(GR.BETA45)
gelman.diag(GR.BETA46)
gelman.diag(GR.BETA47)
gelman.diag(GR.BETA48)

##################
### Prediction ###
##################

time.predict = matrix(NA,nrow = 10000,ncol = 12)
logtime.predict = matrix(NA,nrow = 10000,ncol = 12)
```

```
X.predict = matrix(NA,nrow = 4,ncol = 8)
#X.predict for male (age 39, no children, log(personal income) 10.57) taking bus to work
living in different cities
X.predict[1,]= c(1,0, mean(age),1, mean(aa),1,0,0)
#X.predict for male (age 39, no children, log(personal income) 10.57) taking subway to
work living in different cities
X.predict[2,]= c(1,0, mean(age),1, mean(aa),0,1,0)
#X.predict for male (age 39, no children, log(personal income) 10.57) taking taxi to work
living in different cities
X.predict[3,]= c(1,0, mean(age),1, mean(aa),0,0,1)


for(i in 1:10000)
{
  #group Chi.
  for(j in 1:3)
  {
    logtime.predict[i,j]=t(as.matrix(X.predict[j,]))%*%as.matrix(BETA.post[1,,i+10000])
    time.predict[i,j] = exp(logtime.predict[i,j])
  }

  #group NYC
  for(j in 4:6)
  {
    logtime.predict[i,j]=t(as.matrix(X.predict[j-3,]))%*%as.matrix(BETA.post[2,,i+10000])
    time.predict[i,j] = exp(logtime.predict[i,j])
  }

  #group Phi.
  for(j in 7:9)
  {
    logtime.predict[i,j]=t(as.matrix(X.predict[j-6,]))%*%as.matrix(BETA.post[3,,i+10000])
    time.predict[i,j] = exp(logtime.predict[i,j])
  }

  #group SF.
  for(j in 10:12)
  {
    logtime.predict[i,j]=t(as.matrix(X.predict[j-9,]))%*%as.matrix(BETA.post[4,,i+10000])
    time.predict[i,j] = exp(logtime.predict[i,j])
  }
}

### prediction for taking bus ###

plot(density(logtime.predict[,1]), xlab="predicted logtime", main="Density Plot -
     Predicted Logtime (Male, Bus)", ylim=c(0, 0.33))
lines(density(logtime.predict[,4]), col = "red", lty = 2)
lines(density(logtime.predict[,7]), lty = 3)
lines(density(logtime.predict[,10]), col = "blue", lty = 6)
legend("topright", text.width = 4, x.intersp = 0.5, y.intersp = 0.3, legend=c("Chicago",
"NYC", "Philadelphia", "San Francisco"),
       lty=c(1,2,3,6), col=c("black","red", "black","blue"))

### prediction for taking subway ###

plot(density(logtime.predict[,2]), xlab="predicted logtime", main="Density Plot -
     Predicted Logtime (Male, Subway)", ylim=c(0, 0.4))
lines(density(logtime.predict[,5]), col = "red", lty = 2)
lines(density(logtime.predict[,8]), lty = 3)
lines(density(logtime.predict[,11]), col = "blue", lty = 6)
legend("topright", text.width = 3, x.intersp = 0.5, y.intersp = 0.3, legend=c("Chicago",
"NYC", "Philadelphia", "San Francisco"),
       lty=c(1,2,3,6), col=c("black","red", "black","blue"))

### prediction for taking taxi ###
```

```
plot(density(logtime.predict[,3]), xlab="predicted logtime", main="Density Plot -
     Predicted Logtime (Male, Taxi)", ylim=c(0, 0.25), xlim=c(-7,15))
lines(density(logtime.predict[,6]), col = "red", lty = 2)
lines(density(logtime.predict[,9]), lty = 3)
lines(density(logtime.predict[,12]), col = "blue", lty = 6)
legend("topright", x.intersp = 0.5, y.intersp = 0.3, legend=c("Chicago", "NYC",
"Philadelphia", "San Francisco"),
       lty=c(1,2,3,6), col=c("black","red", "black","blue"))


### prediction for taking all 3 kinds of transportation in Chicago ###

plot(density(logtime.predict[,1]), xlab="predicted logtime", main="Density plot - All 3
Transportations", ylim=c(0, 0.4))
lines(density(logtime.predict[,2]), lty = 2)
lines(density(logtime.predict[,3]), lty = 3)
legend("topright", legend=c("Bus","Subway", "Taxi"), lty=c(1,2,3))

### Boxplot ###

# boxplot of all transportations in all cities
boxplot(time.predict, outline=F, ylab="time", main="Boxplot - All 3 Transportations in 4
Cities")

# boxplot of male taking bus to work
time.predict.wf = matrix(NA, nrow = 10000, ncol = 4)
time.predict.wf[,1] = time.predict[,1]
time.predict.wf[,2] = time.predict[,4]
time.predict.wf[,3] = time.predict[,7]
time.predict.wf[,4] = time.predict[,10]
boxplot(time.predict.wf, xlab = "1=Chicago, 2=NYC, 3=Phi, 4=SF", main = "Boxplot - Male,
Bus", outline = F)

# boxplot of male taking subway to work
time.predict.wf = matrix(NA, nrow = 10000, ncol = 4)
time.predict.wf[,1] = time.predict[,2]
time.predict.wf[,2] = time.predict[,5]
time.predict.wf[,3] = time.predict[,8]
time.predict.wf[,4] = time.predict[,11]
boxplot(time.predict.wf, xlab = "1=Chicago, 2=NYC, 3=Phi, 4=SF", main = "Boxplot - Male,
Subway", outline = F)


# boxplot of male taking taxi to work
time.predict.wf = matrix(NA, nrow = 10000, ncol = 4)
time.predict.wf[,1] = time.predict[,3]
time.predict.wf[,2] = time.predict[,6]
time.predict.wf[,3] = time.predict[,9]
time.predict.wf[,4] = time.predict[,12]
boxplot(time.predict.wf, xlab = "1=Chicago, 2=NYC, 3=Phi, 4=SF", main = "Boxplot - Male,
Taxi", outline = F)


### mean of all predict logtime ###

#logtime.predict.mean = matrix(NA, nrow=1, ncol=12)
#for (i in 1:12)
#{
#  logtime.predict.mean[i] = mean(logtime.predict[,i])
#}
#logtime.predict.mean


#time.predict.mean = exp(logtime.predict.mean)
#time.predict.mean
```

```
#1 chi bus
#2 chi subway
#3 chi taxi
#4 NYC bus
#5 NYC subway
#6 NYC taxi
#7 phi bus
#8 phi subway
#9 phi taxi
#10 sf bus
#11 sf subway
#12 sf taxi

###Prediction for male and female taking subway to work in four cities
time.predict2 = matrix(NA,nrow = 10000,ncol = 8)
logtime.predict2 = matrix(NA,nrow = 10000,ncol = 8)
X.predict2 = matrix(NA,nrow = 4,ncol = 8)
#X.predict for male (age 39, no children, log(personal income) 10.57) taking subway to
work living in different cities
X.predict2[1,]= c(1,0, mean(age),1, mean(aa),0,1,0)
#X.predict for female (age 39, no children, log(personal income) 10.57) taking subway to
work living in different cities
X.predict2[2,]= c(1,0, mean(age),0, mean(aa),0,1,0)


for(i in 1:10000)
{
#group Chi.
      for(j in 1:2)
      {

      logtime.predict2[i,j]=t(as.matrix(X.predict2[j,]))%*%as.matrix(BETA.post[1,,i+10000
])
            time.predict2[i,j] = exp(logtime.predict2[i,j])
      }

#group NYC
      for(j in 3:4)
      {
            logtime.predict2[i,j]=t(as.matrix(X.predict2[j-
2,]))%*%as.matrix(BETA.post[2,,i+10000])
            time.predict2[i,j] = exp(logtime.predict2[i,j])
      }

#group Phi.
      for(j in 5:6)
      {
            logtime.predict2[i,j]=t(as.matrix(X.predict2[j-
4,]))%*%as.matrix(BETA.post[3,,i+10000])
            time.predict2[i,j] = exp(logtime.predict2[i,j])
      }

#group SF.
      for(j in 7:8)
      {
            logtime.predict2[i,j]=t(as.matrix(X.predict2[j-
6,]))%*%as.matrix(BETA.post[4,,i+10000])
            time.predict2[i,j] = exp(logtime.predict2[i,j])
      }
}


boxplot(time.predict2, outline=F, ylab="time", main="Boxplot ? Male and Female taking
subway in 4 Cities")
```

```
###prediction for male having different number of children taking subway to work in four
cities
time.predict3 = matrix(NA,nrow = 10000,ncol = 16)
logtime.predict3 = matrix(NA,nrow = 10000,ncol = 16)
X.predict3 = matrix(NA,nrow = 4,ncol = 8)
#X.predict for male (age 39, no children, log(personal income) 10.57) taking subway to
work living in different cities
X.predict3[1,]= c(1,0, mean(age),1, mean(aa),0,1,0)
#X.predict for male (age 39, 1 child, log(personal income) 10.57) taking subway to work
living in different cities
X.predict3[2,]= c(1,1, mean(age),0, mean(aa),0,1,0)
#X.predict for male (age 39, 2 children, log(personal income) 10.57) taking subway to
work living in different cities
X.predict3[3,]= c(1,2, mean(age),1, mean(aa),0,1,0)
#X.predict for male (age 39, 3 children, log(personal income) 10.57) taking subway to
work living in different cities
X.predict3[4,]= c(1,3, mean(age),0, mean(aa),0,1,0)



for(i in 1:10000)
{
#group Chi.
      for(j in 1:4)
      {

      logtime.predict3[i,j]=t(as.matrix(X.predict3[j,]))%*%as.matrix(BETA.post[1,,i+10000
])
            time.predict3[i,j] = exp(logtime.predict3[i,j])
      }

#group NYC
      for(j in 5:8)
      {
            logtime.predict3[i,j]=t(as.matrix(X.predict3[j-
4,]))%*%as.matrix(BETA.post[2,,i+10000])
            time.predict3[i,j] = exp(logtime.predict3[i,j])
      }

#group Phi.
      for(j in 9:12)
      {
            logtime.predict3[i,j]=t(as.matrix(X.predict3[j-
8,]))%*%as.matrix(BETA.post[3,,i+10000])
            time.predict3[i,j] = exp(logtime.predict3[i,j])
      }

#group SF.
      for(j in 13:16)
      {
            logtime.predict3[i,j]=t(as.matrix(X.predict3[j-
12,]))%*%as.matrix(BETA.post[4,,i+10000])
            time.predict3[i,j] = exp(logtime.predict3[i,j])
      }
}


boxplot(time.predict3, outline=F, ylab="time", main="Boxplot ? Male with 0-4 children
taking subway in 4 Cities")
```