Abstract
This paper includes specific answers to two questions. Part (A) basically discusses the process of computing an estimate, estimate of the variance of an estimator and confidence intervals of the estimator via delta method, normal approximation and non-parametric bootstrap method. Part (B) adopts Hidden Markov Model to estimate X using data Y, and quantities such as the estimated positions, log-likelihood are computed.

Part (A)

Question (a)

To compute the value of estimate $\widehat{\gamma_0}$, we start by calculating $\widehat{\beta_n}$ by using glm() function to evaluate the linear regression model:

$$Y\sim inlf + educ + exper + expersq + othinc + kidslt6 + kisage6$$

We get $\widehat{\beta_n}$=[-3.57622442, 0.23452224, 0.19732315, -0.00363914 -0.02970542, -1.04197703, 0.56866701].

From the definition of $\widehat{\gamma_0}$, we get

$$\widehat{\gamma_0} = \left[\frac{1}{n}\sum_{i=1}^{n}\frac{\partial\varphi(x_i^T*\beta)}{\partial othinc}\right]|\widehat{\beta_n} = \left[\frac{1}{n}\sum_{i=1}^{n}\frac{\partial\{\frac{exp(x_i^T*\beta)}{1+exp(x_i^T*\beta)}\}}{\partial othinc}\right]|\widehat{\beta_n} = \left[\frac{1}{n}\sum_{i=1}^{n}\frac{\beta_5 exp(x_i^T*\beta)}{(1+exp(x_i^T*\beta))\text{^}2}\right]|\widehat{\beta_n}$$

where β_5 is the fifth entry of the vector β. By applying the data, we get $\widehat{\gamma_0} = -0.005698381$.

To get a 95% confidence interval for γ_0, we apply the delta method to get an approximate normal distribution.

By delta method, if g is a function of a statistic T, then

$$g(T)\sim N(g(\mu_T), (g'(\mu_T))^2{\sigma_T}^2)$$

where μ_T is the mean of T and ${\sigma_T}^2$ is the variance of T.
Here, T=β is a multivariate vector, then

$$\gamma_0 = E\{\frac{\partial P(inlf = 1|x)}{\partial othinc}\} = g(\beta)$$

Here we use the maximum likelihood estimate $\widehat{\beta_n}$.

$$g(\widehat{\beta_n}) = \widehat{\gamma_0} = -0.005698381$$

The estimated variance is calculated by using maximum likelihood estimate of cov($\widehat{\beta_n}$) and a jacobian vector $J = \frac{\partial g(X*\widehat{\beta})}{\partial\widehat{\beta}}$ to get the estimated variance of $\widehat{var(\gamma_0)} = J^T **\widehat{cov\beta_n} * J$=2.295148e-06. Then we get a 95% confidence interval

is C=($\widehat{\gamma_0} - sqrt(\widehat{\text{var}(\gamma_0)}) * Z(1-\frac{\alpha}{2})$, $\widehat{\gamma_0} + sqrt(\widehat{\text{var}(\gamma_0)}) * Z(1-\frac{\alpha}{2})$) =(-0.008667677,-0.002729085)

Question (b)

To compute the non-parametric paired bootstrap estimate of $var(\widehat{\gamma_0})$, we adopt the algorithm as the following:

(1) For j=1,2,...B,

Simulate ind = sample(n,replace=TRUE) and simulate bootstrap samples of paired data $((X_1^{*(j)}, Y_1^{*(j)}),(X_2^{*(j)}, Y_2^{*(j)}),...(X_n^{*(j)}, Y_n^{*(j)}))$ from the original data and calculate the $\widehat{\beta_n}^*$ by using glm() function with bootstrap data.Then we Evaluate the $\widehat{\gamma_0}^*$ for each paired data.

(2) Return the bootstrap variance estimate $\sigma_\beta{}^2$=2.52031e-06.

To further calculate the 95% confidence intervals for γ_0, we use the normal and bootstrap methods as the followings:

(1) Normal Method

C = $(\widehat{\gamma_0} - Z_{0.05/2} * \sigma_\beta{}^2, \widehat{\gamma_0} + Z_{0.05/2} * \sigma_\beta{}^2)$ = (-0.008809919 -0.003475833)

(2) Bootstrap Method

C = $(2*\widehat{\gamma_0} - q_{1-0.05/2} * \sigma_\beta{}^2, 2*\widehat{\gamma_0} - q_{0.05/2} * \sigma_\beta{}^2)$ = (-0.008698120 -0.002559321)

Here we find that the confidence intervals via normal and boot methods are similar and both narrower than the confidence interval in (a). As these two methods give more accurate confidence intervals than normal approximation, the bootstrap method is considered to be better when there is no information about the distribution.

Question (c)

Plot the empirical cumulative distribution function of $(\widehat{\gamma_0}^* - \widehat{\gamma_0})$.

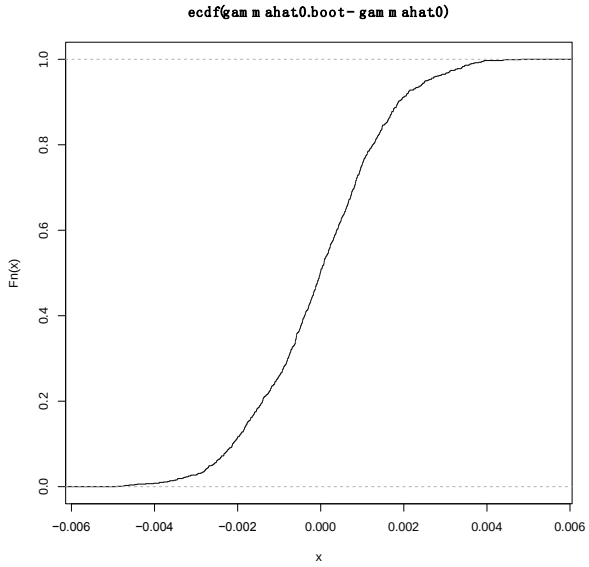

Figure1. Empirical Cumulative Distribution Function

From the figure 1, though the empirical distribution is originally a step function, it is quite smooth for large sample size n=753. As there is no information known about the distribution of $(\widehat{\gamma_0}^* - \widehat{\gamma_0})$, it is a natural estimator. And as n tends to infinity, $F_n(x) \to F(x)$ almost surely, thus we could get a clear understanding about the distribution of $(\widehat{\gamma_0}^* - \widehat{\gamma_0})$ from the graph of empirical distribution.

Question (d)

To consider the marginal effect at the new values $x_0 = (1, 10, 3, 9, 10, 1, 0)^T$, we repeat the procedures in (a), (b).

Firstly, the new $\widehat{\gamma_0}$ is only dependent on data x_0:

$$\widehat{\gamma_0} = \left[\frac{\beta_5 exp(x_0^T * \beta)}{(exp(x_0^T * \beta))\text{^}2}\right] | \widehat{\beta_n} = -0.003093255$$

Similarly, the 95% confidence interval of $\widehat{\gamma_0}$ based on the delta method is given by (-0.0053362789 -0.0008502308).

Secondly, to get the non-parametric paired estimate of $var(\widehat{\gamma_0})$, we adopt the algorithm:

(1) For j=1,2,...B, simulate ind=sample(n,replace=TRUE) and simulate bootstrap samples of paired data $((X_1^{*(j)}, Y_1^{*(j)}),(X_2^{*(j)}, Y_2^{*(j)}),...(X_n^{*(j)}, Y_n^{*(j)}))$ from the original data and calculate $\widehat{\beta_n}^*$, and then evaluate the $\widehat{\gamma_0}^*$ not for each paired data, but all for the same data x_0.

(2) Return the bootstrap variance estimate 1.521406e-06.

And we further get the 95% confidence intervals for normal method and bootstrap

method respectively as

$C = (\widehat{\gamma_0} - Z_{0.05/2} * {\sigma_\beta}^2,\ \widehat{\gamma_0} + Z_{0.05/2} * {\sigma_\beta}^2) = (-0.005510778\ -0.001366436)$

$C=(2*\ \widehat{\gamma_0} - q_{1-0.05/2} * {\sigma_\beta}^2\ ,2*\ \widehat{\gamma_0} - q_{0.05/2} * {\sigma_\beta}^2\ ) = (-0.0051460379\ -0.0003200011)$

From these results, compared to the average marginal effect, this marginal effect only considers the data x0 which makes less use of the data x in the question. Therefore, it might generate estimates and confidence intervals as accurate as the average effect does.

Question (e)

Now we are interested in the marginal effect on the probability of being in the labor force of having a child less than 6 years old with the same data x_0. Thus we adopt the same method as before with a new definition of $\gamma_0, \widehat{\gamma_0}$:

$$\widehat{\gamma_0} = \frac{\partial \varphi(x_0^T * \beta)}{\partial kidslt6} |\widehat{\beta_n} = \left[\frac{\beta_6 exp(x_0^T * \beta)}{(exp(x_0^T * \beta))\text{^}2}\right] |\widehat{\beta_n} = -0.1085021$$

And by applying the similar procedures as above, we get the 95% bootstrap confidence interval as (-0.1438993 -0.0698345). This shows that if we do repeated experiments, there are 95% of the times that the true value is within this range. And compared to result in (d), the marginal effect of this variable is more significant than the effect of othinc.

Part (B)

Question (a) (i)

The linear Gaussian state-space model is defined as for t = 1,2,3,...T:

$$X_t = F_t * X_{t-1} + G_t * V_t$$
$$Y_t = H_t * X_t + W_t$$

where $X_t = \begin{pmatrix} P_t^x & P_t^y & V_t^x & V_t^y \end{pmatrix}^T$, $Y_t = \begin{pmatrix} P_t^x & P_t^y \end{pmatrix}^T + W_t$

$$F_t = \begin{pmatrix} 1 & 0 & \delta & 0 \\ 0 & 1 & 0 & \delta \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},\ G_t = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},\ V_t = \begin{pmatrix} V_t^x \\ V_t^y \end{pmatrix},\ H_t = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Here Y is the GPS measurements of the position of a moving car; X is the position and velocity vector.

Question (a) (ii)

For t=1,2...,T (T=90), the filtering mean $\mu_{t|t}$ and the covariance matrix $\Sigma_{t|t}$ are computed using the kalman function as:

Here only the filtering mean and covariance matrix for t=1,2,3 are shown, quantities for other times are computed in Appendix.

The filtering mean:

```
           [,1]       [,2]      [,3]
[1,] -4.520923 -14.376082 -6.656538
[2,]  7.116699   2.046364 -2.760024
[3,]  0.000000   0.000000  0.000000
[4,]  0.000000   0.000000  0.000000
```

The covariance matrix:

```
          [,1]     [,2] [,3] [,4]
[1,] 221.0216   0.0000    0    0
[2,]   0.0000 221.0216    0    0
[3,]   0.0000   0.0000  100    0
[4,]   0.0000   0.0000    0  100

, , 2

          [,1]     [,2] [,3] [,4]
[1,] 207.8158   0.0000    0    0
[2,]   0.0000 207.8158    0    0
[3,]   0.0000   0.0000  100    0
[4,]   0.0000   0.0000    0  100

, , 3

          [,1]     [,2] [,3] [,4]
[1,] 207.7384   0.0000    0    0
[2,]   0.0000 207.7384    0    0
[3,]   0.0000   0.0000  100    0
[4,]   0.0000   0.0000    0  100
```

Question (a)(iii)

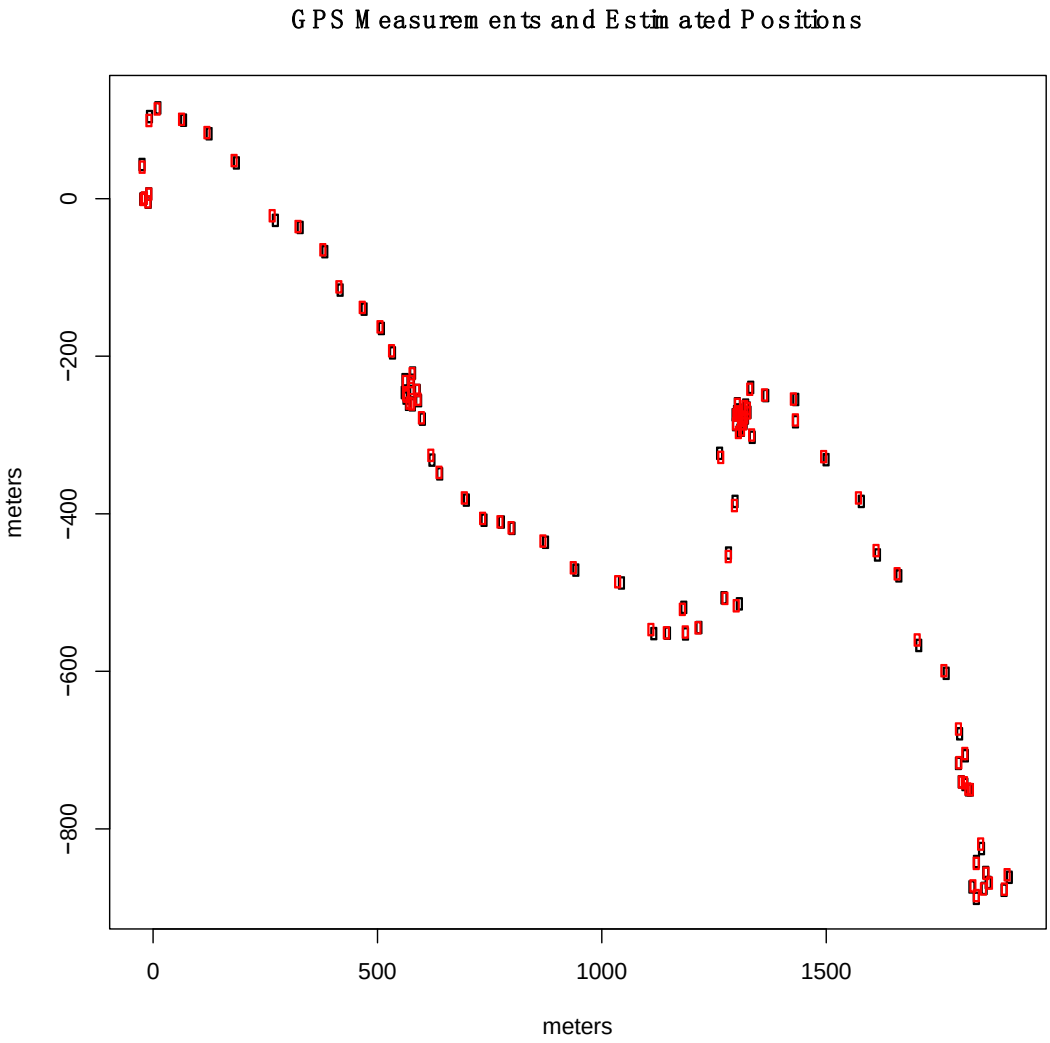

Figure2. Estimated positions and GPS observations

In fig2, the red points in the graph are the estimated positions and the black points and the GPS observations. From the distribution of estimated positions, it is estimated that from time t=1 to T, the car is at approximately the same position as the observations detected from GPS, which implies the negligible GPS error W_t.

Question (a)(iv)

The uncertainty on the position of the car is represented by a confidence interval in the graph.

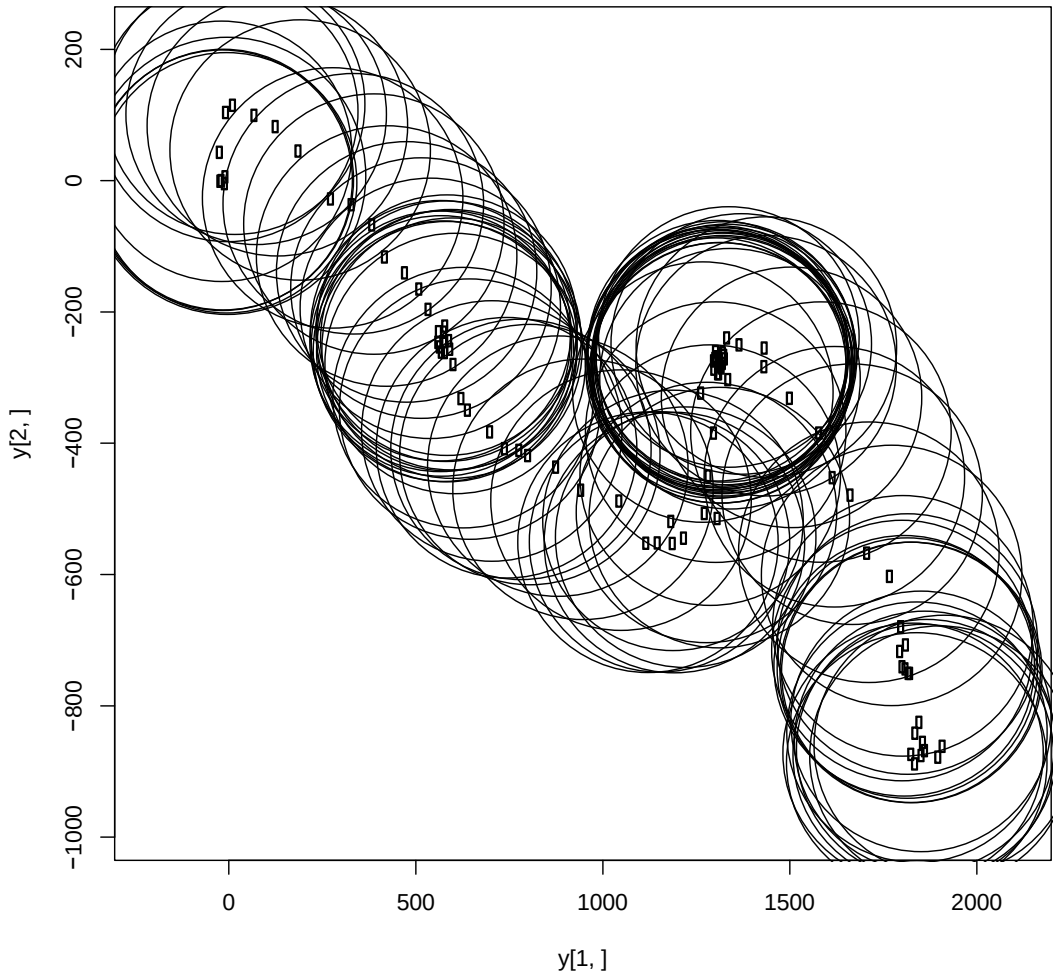

Figure3. Uncertainty on the position of the car in the x and y coordinate

In Figure 3, the little boxes represent the positions of moving car in x-coordinate and y-coordinate and the circles show the 95% confidence intervals of its position for each time from 1 to T from an approximate normal distribution . And this gives the range of uncertainty of the position of the car.

Question (b)(i)

Compared to the original kalman function, we add two quantities to compute the log-likelihood of y.

As we aim to maximize the log-likelihood and

$$\log p(y_1, y_2, \dots y_T) = \log p(y_1) + \sum_{t=2}^{T} \log p(y_t | y_1, y_2, \dots y_{t-1})$$

Where $p(y_1)$ and $p(y_t | y_1, y_2, \dots y_{t-1})$ have Gaussian distributions such that

$$p(y_1) \sim N(y_1; \widehat{y_{1|0}}, S_1)$$

$$p(y_t|y_1, y_2, \ldots y_{t-1}) \sim N(y_t; \widehat{y_{t|t-1}}, S_t)$$

Then we only need to add one more variable $\log p(y)$ which is a vector with entries $\log(p(y_1)), \log(p(y_2)),\ldots\log(p(y_T))$ and output the sum of all entries of this vector to get the log-likelihood.The modified R function kalman is shown in the Appendix.

Question (b)(ii)

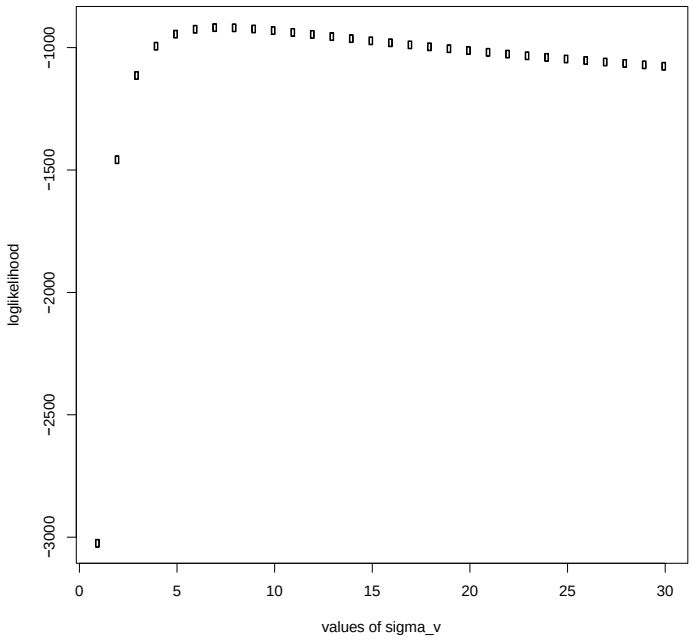

Figure4. Loglikelihood against the values of sigma_v

By computing the log-likelihood for $\sigma_v = 1, 2, \ldots 100$, we get the maximum likelihood estimate of σ_v=7. And from fig 4, it is clear that the log-likelihood first increases greatly from σ_v=0 to σ_v=7, and then it slowly decreases when σ_v increases from 7. Therefore, the maximum likelihood estimate of σ_v is 7.

Appendix
Part A

```
library(numDeriv)
require(wooldridge)
inlf = mroz$inlf
educ = mroz$educ
exper = mroz$exper
expersq = mroz$expersq
othinc = mroz$nwifeinc
kidslt6 = 1*(mroz$kidslt6>0)
kidsge6 = 1*(mroz$kidsge6>0)
mdata = data.frame(inlf,educ,exper,expersq,othinc,kidslt6,kidsge6)
logitres = glm(inlf ~ educ + exper + expersq + othinc
                        + kidslt6 + kidsge6, data = mdata, family = "binomial")
## Question A(a)
# Define beta.hat.n
beta.hat.n = numeric(7)
for (i in 1:7){
   beta.hat.n[i]=logitres$coefficients[i]
}
# Generate data vector x(i)
x <- function(i){
   a=numeric(7)
   a[1]=1
   a[2]=educ[i]
   a[3]=exper[i]
   a[4]=expersq[i]
   a[5]=othinc[i]
   a[6]=kidslt6[i]
   a[7]=kidsge6[i]
   return(a)
}
# Calculate Gamma_hat_0
n=753
b = numeric(n)
c = numeric(n)
for (i in 1:n){
   c[i]=exp(x(i) %*% beta.hat.n)
   b[i]=c[i]/(1+c[i])^2*beta.hat.n[5]/n
}
gammahat.0=sum(b)
print(gammahat.0)
## [1] -0.005698381
# CI using normal approximation
```

```
alpha=0.05
var=vcov(logitres)
h<-function(beta){
   for (i in 1:n){
      c[i]=exp(x(i) %*% beta)
      b[i]=c[i]/(1+c[i])^2*beta[5]/n
   }
   sum(b)
}
J=jacobian(h,beta.hat.n)
var.delta=J%*%var%*%t(J)
# Calculate the confidence interval
ci.normal.gamma.0=c(gammahat.0-sqrt(var.delta)*qnorm(1-
alpha/2),gammahat.0+sqrt(var.delta)*qnorm(1-alpha/2))
print(ci.normal.gamma.0)
## [1] -0.008667677 -0.002729085
## Question (b)
# Non-parametric paired bootstrap estimate
bootstrap_variance_gammahat.0<-function(B){
   beta.hat.n.boot=numeric(7)
   gammahat.0.boot=numeric(B)
   for (j in 1:B) {
      ind = sample(n,replace=TRUE)
      mdata.boot <- mdata[ind,]
      logitres.boot = glm(inlf ~ educ + exper + expersq + othinc
                          + kidslt6 + kidsge6, data = mdata.boot, family =
"binomial")
      for (i in 1:7){
      beta.hat.n.boot [i]= logitres.boot$coefficients[i]
   }
      for (i in 1:n){
         c[i]=exp(x(i) %*% beta.hat.n.boot)
         b[i]=c[i]/(1+c[i])^2*beta.hat.n.boot[5]/n
      }
      gammahat.0.boot[j]=sum(b)
   }
   return(c(var(gammahat.0.boot),gammahat.0.boot))
}
var.gammahat.0.boot=bootstrap_variance_gammahat.0(1000)[1]
gammahat.0.boot=bootstrap_variance_gammahat.0(1000)[2:1000+1]
# CI using normal method
ci.bootnormal.gamma.0=c(gammahat.0-sqrt(var.gammahat.0.boot)*qnorm(1-
alpha/2),gammahat.0+sqrt(var.gammahat.0.boot*qnorm(1-alpha/2)))
print(ci.bootnormal.gamma.0)
```

```
## [1] -0.009020062 -0.003325729
# CI using bootstrap method
ci.bootstrap.gamma.0=c(2*gammahat.0-quantile(gammahat.0.boot,1-
alpha/2,names=FALSE),2*gammahat.0-
quantile(gammahat.0.boot,alpha/2,names=FALSE))
print(ci.bootstrap.gamma.0)
## [1] -0.008916743 -0.002559069
# (c) Plot the empricial cdf
pdf("Empirical cdf.pdf", height=8, width=8)
plot(ecdf(gammahat.0.boot-gammahat.0))
dev.off()
## quartz_off_screen
##                             2
# (d) Marginal Effect at x0
x0=c(1,10,3,9,10,1,0)
# Calculate new Gamma_hat_0
r=exp(x0 %*% beta.hat.n)
new.gammahat.0=beta.hat.n[5]/(1+r)^2*r
print(new.gammahat.0)
##                          [,1]
## [1,] -0.003093255
# CI using normal approximation
g<-function(beta){
      c=exp(x0 %*% beta)
      b=c/(1+c)^2*beta[5]
}
new.J=jacobian(g,beta.hat.n)
new.var.delta=new.J%*%var%*%t(new.J)
new.ci.normal.gamma.0=c(new.gammahat.0-sqrt(new.var.delta)*qnorm(1-
alpha/2),new.gammahat.0+sqrt(new.var.delta)*qnorm(1-alpha/2))
print(new.ci.normal.gamma.0)
## [1] -0.0053362789 -0.0008502308
# nonparametric paired bootstrap estimate using x0
new_bootstrap_variance_gammahat.0<-function(B){
   beta.hat.n.boot=numeric(7)
   gammahat.0.boot=numeric(B)
   for (j in 1:B) {
      ind = sample(n,replace=TRUE)
      mdata.boot <- mdata[ind,]
      logitres.boot = glm(inlf ~ educ + exper + expersq + othinc
                          + kidslt6 + kidsge6, data = mdata.boot, family =
"binomial")
      for (i in 1:7){
         beta.hat.n.boot[i]=logitres.boot$coefficients[i]
```

```
    }
    c=exp(x0 %*% beta.hat.n.boot)
    gammahat.0.boot[j]=c/(1+c)^2*beta.hat.n.boot[5]
  }
  return(c(var(gammahat.0.boot),gammahat.0.boot))
}
new.var.gammahat.0.boot=new_bootstrap_variance_gammahat.0(1000)[1]
new.gammahat.0.boot=new_bootstrap_variance_gammahat.0(1000)[2:1000+1]
# CI using normal method
new.ci.bootnormal.gamma.0=c(new.gammahat.0-
sqrt(new.var.gammahat.0.boot)*qnorm(1-
alpha/2),new.gammahat.0+sqrt(new.var.gammahat.0.boot*qnorm(1-alpha/2)))
print(new.ci.bootnormal.gamma.0)
## [1] -0.005525012 -0.001356269
# CI using bootstrap method
new.ci.bootstrap.gamma.0=c(2*new.gammahat.0-
quantile(new.gammahat.0.boot,1-alpha/2,names=FALSE),2*new.gammahat.0-
quantile(new.gammahat.0.boot,alpha/2,names=FALSE))
print(new.ci.bootstrap.gamma.0)
## [1] -0.0049901174 -0.0002938789
# (e) Calculate new Gamma_hat_0
r=exp(x0 %*% beta.hat.n)
e.new.gammahat.0=beta.hat.n[6]/(1+r)^2*r
print(e.new.gammahat.0)
##                    [,1]
## [1,] -0.1085021
# nonparametric paired bootstrap estimate using x0
e_new_bootstrap_variance_gammahat.0<-function(B){
  beta.hat.n.boot=numeric(7)
  gammahat.0.boot=numeric(B)
  for (j in 1:B) {
    ind = sample(n,replace=TRUE)
    mdata.boot <- mdata[ind,]
    logitres.boot = glm(inlf ~ educ + exper + expersq + othinc
                        + kidslt6 + kidsge6, data = mdata.boot, family =
"binomial")
    for (i in 1:7){
      beta.hat.n.boot[i]=logitres.boot$coefficients[i]
    }
    c=exp(x0 %*% beta.hat.n.boot)
    gammahat.0.boot[j]=c/(1+c)^2*beta.hat.n.boot[6]
  }
  return(c(var(gammahat.0.boot),gammahat.0.boot))
}
```

```
e.new.gammahat.0.boot=e_new_bootstrap_variance_gammahat.0(1000)[2:1000
+1]
# CI using bootstrap method
e.new.ci.bootstrap.gamma.0=c(2*e.new.gammahat.0-
quantile(e.new.gammahat.0.boot,1-
alpha/2,names=FALSE),2*e.new.gammahat.0-
quantile(e.new.gammahat.0.boot,alpha/2,names=FALSE))
print(e.new.ci.bootstrap.gamma.0)
## [1] -0.14491822 -0.06811068
```

Part(B)

```
library("mvtnorm")
y=data.matrix(read.csv("observations.csv",header=FALSE))
kalman = function(y, F, G, Q, H, R, mu0, Sigma0) {
dy = nrow(y)
T = ncol(y)
dx = length(mu0)
I = diag(dx)
## INITIALIZATION ##
mu.p = matrix(0, nrow = dx, ncol = T)
Sigma.p = array(0, c(dx, dx, T))
mu.f = matrix(0, nrow = dx, ncol = T)
Sigma.f = array(0, c(dx, dx, T))
mu.s = matrix(0, nrow = dx, ncol = T)
Sigma.s = array(0, c(dx, dx, T))
log.py = numeric(T)
mu.p[, 1] = F %*% mu0
Sigma.p[, , 1] = F %*% Sigma0 %*% t(F) + G %*% Q %*% t(G)
nu = y[, 1] - H %*% mu.p[, 1]
S = H %*% Sigma.p[, , 1] %*% t(H) + R
K = Sigma.p[, , 1] %*% t(H) %*% solve(S)
mu.f[, 1] = mu.p[, 1] + K %*% nu
Sigma.f[, , 1] = (I - K %*% H) %*% Sigma.p[, , 1]
log.py[1] = dmvnorm(y[, 1], mean=H %*% mu.p[, 1],sigma=S, log=TRUE)
for (t in (2:T)) {
mu.p[, t] = F %*% mu.f[, t - 1]
Sigma.p[, , t] = F %*% Sigma.f[, , t - 1] %*% t(F) + G %*% Q %*% t(G)
nu = y[, t] - H %*% mu.p[, t]
S = H %*% Sigma.p[, , t] %*% t(H) + R
K = Sigma.p[, , t] %*% t(H) %*% solve(S)
mu.f[, t] = mu.p[, t] + K %*% nu
Sigma.f[, , t] = (I - K %*% H) %*% Sigma.p[, , t]
log.py[t]=dmvnorm(y[, t], mean=H %*% mu.p[, t],sigma=S, log=TRUE)
}
```

```
mu.s[, T] = mu.f[, T]
Sigma.s[, , T] = Sigma.f[, , T]
for (t in (T - 1):1) {
J = Sigma.f[, , t] %*% t(F) %*% solve(Sigma.p[, , t + 1])
mu.s[, t] = mu.f[, t] + J %*% (mu.s[, t + 1] - mu.p[, t + 1])
Sigma.s[, , t] = Sigma.f[, , t] + J %*% (Sigma.s[, , t + 1] - Sigma.p[, , t + 1]) %*% t(J)
}
sum=sum(log.py)
return(list(mu.f = mu.f, Sigma.f = Sigma.f, mu.p = mu.p, Sigma.p = Sigma.p,
                    mu.s = mu.s, Sigma.s = Sigma.s, sum = sum))
}

# Question (a)(ii)
v=10
R = diag(15^2,2)
mu0 = matrix(0,nrow=4,ncol=1)
Sigma0 =
matrix(c(100^2,0,0,0,0,100^2,0,0,0,0,10^2,0,0,0,0,10^2),nrow=4,ncol=4,byrow
=TRUE)
G = matrix(c(0,0,0,0,1,0,0,1),nrow=4,ncol=2,byrow=TRUE)
Q = diag(v^2,2)
H = matrix(c(1,0,0,0,0,1,0,0),nrow=2,ncol=4,byrow=TRUE)
F = matrix(c(1,0,5,0,0,1,0,5,0,0,0,0,0,0,0,0),nrow=4,ncol=4,byrow=TRUE)
results.KF = kalman(y, F, G, Q, H, R, mu0, Sigma0)
mu.f = results.KF$mu.f
Sigma.f = results.KF$Sigma.f
print(mu.f)
##                [,1]           [,2]          [,3]             [,4]          [,5]             [,6]
[,7]
## [1,] -4.520923 -14.376082 -6.656538 -17.3162310 -19.37493    -4.123704
14.13573
##  [2,]      7.116699         2.046364   -2.760024         0.3939777      41.29530
100.491113 115.18870
##  [3,]      0.000000         0.000000      0.000000         0.0000000         0.00000
0.000000      0.00000
##  [4,]      0.000000         0.000000      0.000000         0.0000000         0.00000
0.000000      0.000
# Question (a)(iii)
pdf("Estimated Positions.pdf", height=8, width=8)
plot(y[1,],y[2,], xlab="meters", ylab="meters",main="GPS Measurements and
Estimated Positions",col=1)
points(mu.f[1,],mu.f[2,],col=2)
dev.off()
# Question (a)(iv)
```

```
pdf("Uncertainty.pdf", height=8, width=8)
plot(y[1,],y[2,],xlab="uncertainty in x coordinate",ylab="uncertainty in y coordinate")
points
## function (x, ...)
## UseMethod("points")
## <bytecode: 0x7fb3683eea10>
## <environment: namespace:graphics>
dev.off()
# Question (b)(ii)
res = numeric(30)
for (i in 1:30){
   Q = diag(i^2,2)
   results.KF.i = kalman(y, F, G, Q, H, R, mu0, Sigma0)
   res[i]=results.KF.i$sum
}
which.max(res)
## [1] 7
pdf("loglikelihood.pdf", height=8, width=8)
plot(1:30,res,xlab="values of sigma_v",ylab="loglikelihood")
points
## function (x, ...)
## UseMethod("points")
## <bytecode: 0x7fb3683eea10>
## <environment: namespace:graphics>
dev.off()
```