Data=read.csv("E:ensyu\_cleaningallダミー.csv",header=T)

## データ数:Data の行数を数える

hh <- nrow(Data)

## パラメータの初期値の設定

b0 <- numeric(9)

##### Logit model の対数尤度関数の定義 #####

fr <- function(x) {

### パラメータの宣言：

## 定数項

b1 <- x[1]

b2 <- x[2]

b3 <- x[3]

b4 <- x[4]

## 目的地までの所要時間

d1 <- x[5]

#d2 <- x[]

#d3 <- x[]

#d4 <- x[]

#d5 <- x[]

## 料金

f1 <-x[6]

##雨ダミー

r1 <-x[7]

##晴ダミー

s1 <-x[8]

##休日ダミー

h1 <-x[9]

## 対数尤度のための変数を宣言

LL = 0

### 今回用いる目的地は以下の 5 つ．

## 鉄道(train)

## バス(bus)

## 自動車(car)

## 自転車(bike)

## 徒歩(walk)

## 効用の計算:説明変数にしたい列を入れる．

# 時間 # 料金

# 定数項

train <-Data$代替手段生成可否train \* exp(d1\*(Data$総所要時間train/100) + b1\*matrix(1,nrow =hh,ncol=1)+r1\*Data$雨ダミー2)

bus <- Data$代替手段生成可否bus \* exp(d1\*(Data$総所要時間bus/100)+ f1\*(Data$費用train/100) + b2\*matrix(1,nrow =hh,ncol=1))

car <- Data$代替手段生成可否car \* exp(d1\*(Data$所要時間car/100) + f1\*(Data$費用train/100) + b3\*matrix(1,nrow =hh,ncol=1)+r1\*Data$雨ダミー2)

bike <- Data$代替手段生成可否bike \* exp(d1\*(Data$所要時間bike/100)+ b4\*matrix(1,nrow =hh,ncol=1)+s1\*Data$晴ダミー2)

walk <-Data$代替手段生成可否walk \* exp(d1\*(Data$所要時間walk/100)+s1\*Data$晴ダミー2+h1\*Data$休日ダミー)

### 選択確率の計算

## 分母となる，各々の exp(V)の和をつくる

a<- exp(car) + exp(train) + exp(bike) + exp(walk) + exp(bus)

## それぞれ計算する

Ptrain <-exp(train)/a

Pbus <- exp(bus)/a

Pcar <-exp(car)/a

Pbike <-exp(bike)/a

Pwalk <-exp(walk)/a

## 選択確率が 0 になってしまった場合に起こる問題の回避

Ptrain <- (Ptrain!=0) \* Ptrain + (Ptrain ==0)

Pbus <- (Pbus!=0) \* Pbus + (Pbus ==0)

Pcar <- (Pcar!=0) \* Pcar + (Pcar ==0)

Pbike <- (Pbike!=0) \* Pbike + (Pbike ==0)

Pwalk <- (Pwalk!=0) \* Pwalk + (Pwalk ==0)

## 選択結果

Ctrain <- Data$代表交通手段 == "鉄道"

Cbus <- Data$代表交通手段 == "バス"

Ccar <- Data$代表交通手段 == "自動車"

Cbike <- Data$代表交通手段 == "自転車"

Cwalk <- Data$代表交通手段 == "徒歩"

## 対数尤度の計算

LL <- colSums(Ctrain\*log(Ptrain) + Ccar \*log(Pcar) + Cbike \*log(Pbike) +Cwalk \*log(Pwalk) + Cbus \*log(Pbus))

}

##### 対数尤度関数 fr の最大化#####

##パラメータ値の最適化

res <- optim(b0,fr, method = "BFGS", hessian = TRUE, control=list(fnscale=-1))

## パラメータ推定値，ヘッセ行列

b <- res$par

hhh <- res$hessian

## t 値の計算

tval <- b/sqrt(-diag(solve(hhh)))

## 初期尤度

L0 <- fr(b0)

## 最終尤度

LL <- res$value

##### 結果の出力 #####

print(res)

## 初期尤度

print(L0)

## 最終尤度

print(LL)

##ρ^2 値

print((L0-LL)/L0)

## 修正済ρ^2 値

print((L0-(LL-length(b)))/L0)

##パラメータ推定値

print(b)

## t 値

print(tval)