# Cancer_Screening_Project_V2

## Overview

Risk factor variables were identified in a previous research (Lee2013). These are:

- (Bi) Static Variables (one time measure):
  - Age
  - Ethnicity
  - Smoker
  - Alkaline Phosphatase
  - Blood platelets
  - Esophageal varices
- (SDi) Standard Deviation of patients' AFP Readings
- (RRi) Least Squares estimate for the rate of rise over time amongst patient's AFP Readings

The statistics of these variables are described in Lee2013 and in Phd dissertation (For more information about the selection of features and sampling methods, Refer to Cancer_Screening_ProjectV1 summary or the Phd dissertation chapter II).

## Data Simulation

Data will be generated based on information from Table 2.1, and Figures 2.1,2.2,2.3.

There are N=82 HCC individuals and N=885 No-HCC individuals in the whole population (For more infor refer to V1 summary)

The sampled data contained a ratio of 1:3 HCC vs non-HCC. So the sample must contain 82 HCC cases and 246 non HCC controls (For more information about the selection of features and sampling methods, Refer to Cancer_Screening_ProjectV1 summary or the Phd dissertation chapter II).

We want to generate simulated data observation for each individuals (Statistics are based on Table 2.1)

### Static Data

#### Age at Baseline

HCC Individuals: 53+/-7

```
HCC_Age <- trunc(rnorm(82, mean = 53, sd = 7))
```

NonHCC Individuals: 50+/-7

```
NonHCC_Age <- trunc(rnorm(246, mean = 50, sd = 7))
```

#### Ethnicity

HCC Individuals: * White = 64% = 53 (1) * Black = 24% = 20 (2) * Hispanic = 6% = 5 (3) * Other = 5% = 4 (4)

```
HCC_ethnicity <- c(rep("1", 53), rep("2", 20), rep("3", 5), rep("4", 4))
```

NonHCC Individuals: * White = 73% = 179 (1) * Black = 18% = 44 (2) * Hispanic = 8% = 19 (3) * Other = 2% = 4 (4)

```
NonHCC_ethnicity <- c(rep("1", 179), rep("2", 44), rep("3", 19), rep("4", 4))
```

### Smoker

HCC Individuals: * Yes = 83% = 68 (1) * No = 17% = 14 (0)

```
HCC_Smoker <- c(rep("1", 68), rep("0", 14))
```

NonHCC Individuals: * Yes = 74% = 182 (1) * No = 26% = 64 (0)

```
NonHCC_Smoker <- c(rep("1", 182), rep("0", 64))
```

### Alkaline Phosphatase

HCC Individuals: 117+/-59

```
HCC_Alk <- trunc(rnorm(82, mean = 117, sd = 59))
```

NonHCC Individuals: 97+/-43

```
NonHCC_Alk <- trunc(rnorm(246, mean = 97, sd = 43))
```

### Blood platelets

HCC Individuals: 126+/-51

```
HCC_platelets <- trunc(rnorm(82, mean = 126, sd = 51))
```

NonHCC Individuals: 97+/-43

```
NonHCC_platelets <- trunc(rnorm(246, mean = 97, sd = 43))
```

### Esophageal varices

HCC Individuals: * Yes = 41% = 34 (1) * No = 59% = 48 (0)

```
HCC_esoph <- c(rep("1", 34), rep("0", 48))
```

NonHCC Individuals: * Yes = 24% = 59 (1) * No = 76% = 187 (0)

```
NonHCC_esoph <- c(rep("1", 59), rep("0", 187))
```

### Data Frame of Patients' Static Data

The first 82 rows belongs to HCC cases and the rest 246 to non HCC.

```
Patients_StaticData <- data.frame(P_ID = c(1:328),
           Age = c(HCC_Age, NonHCC_Age),
           Ethnicity = c(HCC_ethnicity, NonHCC_ethnicity),
           Smoker = c(HCC_Smoker, NonHCC_Smoker),
           Alkaline_Phosphate = c(HCC_Alk, NonHCC_Alk),
           Blood_Platelets = c(HCC_platelets, NonHCC_platelets),
           Esophaegal_Varices = c(HCC_esoph, NonHCC_esoph),
           HCC = c(rep("1", 82), rep("0", 246)))
```

Below is a summary of the static data. Each variable should follow the distribution stated above (similar to Table 2.1)

```
summary(Patients_StaticData)
```

```
##       P_ID             Age        Ethnicity Smoker  Alkaline_Phosphate
##  Min.   :  1.00   Min.   :29.00   1:232     0: 78   Min.   :-63.00
```

```
##  1st Qu.: 82.75   1st Qu.:45.00   2: 64     1:250   1st Qu.: 62.75
##  Median :164.50   Median :50.00   3: 24             Median : 99.50
##  Mean   :164.50   Mean   :50.34   4:  8             Mean   : 97.53
##  3rd Qu.:246.25   3rd Qu.:55.00                     3rd Qu.:130.25
##  Max.   :328.00   Max.   :70.00                     Max.   :287.00
##  Blood_Platelets Esophaegal_Varices HCC
##  Min.   :-12.0   0:235              0:246
##  1st Qu.: 75.0   1: 93              1: 82
##  Median :100.0
##  Mean   :103.1
##  3rd Qu.:137.2
##  Max.   :232.0
```

```
Patients_StaticData$HCC <-  relevel(Patients_StaticData$HCC, "1")
```

## Non-Static Data (AFP Measures) - SKIPPEPD

It has been proven that AFP rate of rise and standard deviation are important measures affecting HCC (Refer to Lee2013)

- In the first 3.5 years: a screening is done every 3 months. So each individual has for sure 13 AFP screenings (time t=0 to t=13) **unless** patient has HCC before time t=13.
- During the screening period (from enrollment to HCC diagnosis or end of follow up), subjects had a median of 18 (range 5-23) AFP tests performed (refer to phd dissertation)

We want to generate the number of screenings each patient receives. We know that 50% of the 328 (82+246) patients did less than 18 screening and 50% did more.

Below we generate a vector containing the number of screening per patient t:

```
screening_numb <- c(sample(5:18, 164, replace = T), sample(18:23, 164, replace = T))
```

*Alternatively:* The number of screenings could be generated using the follow up time (days) distribution of each of the HCC vs. NonHCC group (follow up time is provided in table 2.1)

### AFP Rate of Rise

HCC Individuals: 5+/-11

```
#HCC_AFP_RR <- trunc(rnorm(82, mean = 5, sd = 11))
```

- Each element in the HCC_AFP_RR vector represents the rate of rise of an HCC patient throughout the trial (at the end of trial) and;

### AFP Standard Deviation

HCC Individuals: 51+/-86 For each patient, I will generate t amount of data representing the AFP measure which follows the distribution above and rate of rise specified by HCC_AFP_RR

# Data Generation As per Prof. Mucahit Instructions

## Static Data

I will take N=50 (patients) - k will be 10 (10 patients each iteration) The N patients will be taken from the dataframe already generated above (15 HCC and 35 nonHCC)

```
yes <- (Patients_StaticData[which(Patients_StaticData$HCC=="1"),])
no <- (Patients_StaticData[which(Patients_StaticData$HCC=="0"),])
Patients_StaticData2 <- rbind(yes[sample(1:nrow(yes), 15),], no[sample(1:nrow(no), 35),])
#write.csv(Patients_StaticData2,"Patient_StaticData2.csv")
Patients_StaticData2 <- Patients_StaticData2[,-1]
```

## AFP Data

Generate 1,000 AFP measure. I randomly generate the measures with a mean 90ng/ml and sd 4.

```
AFP <- trunc(rnorm(1000, mean = 90, sd = 4))
AFP_measures <- data.frame(Time = rep(1:10, 100), AFP=AFP)
```

Create standard deviation:

```
mysdfunc <- function(x){
  standard_Dev <- c()
  vec <- c()
  for (i in 1:length(x)){
    vec[i] <- x[i]
    standard_Dev[i] <- sd(vec)
  }
  return(standard_Dev)
}

n <- 100 ## number of chunks
dfchunk <- split(AFP_measures, factor(sort(rank(row.names(AFP_measures))%%n)))
dfchunk[1]
```

```
## $`0`
##    Time AFP
## 1     1  95
## 2     2  88
## 3     3  86
## 4     4  93
## 5     5  92
## 6     6  92
## 7     7  94
## 8     8  89
## 9     9  91
## 10   10  81
```

```
as.data.frame(dfchunk[1])[,2]
```

```
##  [1] 95 88 86 93 92 92 94 89 91 81
```

```
mysdfunc(as.data.frame(dfchunk[1])[,2])
```

```
##  [1]       NA 4.949747 4.725816 4.203173 3.701351 3.346640 3.258688
```

```
##  [8] 3.136764 2.934469 4.228212
```

```
mylist <- list()

for(i in 1:100){
  mylist[[i]] <- mysdfunc(as.data.frame(dfchunk[i])[,2])
}

AFP_measures$SD <- unlist(mylist)
AFP_measures[which(is.na(AFP_measures$SD)),]$SD <- 0
head(AFP_measures,13)
```

```
##    Time AFP       SD
## 1     1  95 0.000000
## 2     2  88 4.949747
## 3     3  86 4.725816
## 4     4  93 4.203173
## 5     5  92 3.701351
## 6     6  92 3.346640
## 7     7  94 3.258688
## 8     8  89 3.136764
## 9     9  91 2.934469
## 10   10  81 4.228212
## 11    1  84 0.000000
## 12    2  84 0.000000
## 13    3  90 3.464102
```

Create Rate of Rise There is a specific equation for this calcuation (cited in p.12 of phd dissertation). For simplicity, we use a standard rate of change measure: (AFP Final - AFP Baseline)/(AFP Baseline)

```
#(AFP_measures$AFP[1:10]-AFP_measures$AFP[1])/AFP_measures$AFP[1]

rateofrise <- function(x){
  return((x-x[1])/x[1])
}

rateofrise(as.data.frame(dfchunk[1])[,2])
```

```
##  [1]  0.00000000 -0.07368421 -0.09473684 -0.02105263 -0.03157895
##  [6] -0.03157895 -0.01052632 -0.06315789 -0.04210526 -0.14736842
```

```
mylist <- list()

for(i in 1:100){
  mylist[[i]] <- rateofrise(as.data.frame(dfchunk[i])[,2])
}

AFP_measures$RR <- unlist(mylist)
head(AFP_measures,13)
```

```
##    Time AFP       SD          RR
## 1     1  95 0.000000  0.00000000
## 2     2  88 4.949747 -0.07368421
## 3     3  86 4.725816 -0.09473684
## 4     4  93 4.203173 -0.02105263
## 5     5  92 3.701351 -0.03157895
## 6     6  92 3.346640 -0.03157895
```

```
## 7     7  94 3.258688 -0.01052632
## 8     8  89 3.136764 -0.06315789
## 9     9  91 2.934469 -0.04210526
## 10   10  81 4.228212 -0.14736842
## 11    1  84 0.000000  0.00000000
## 12    2  84 0.000000  0.00000000
## 13    3  90 3.464102  0.07142857
```

Final AFP dataframe include: Time, AFP, SD, RR

```
#write.csv(AFP_measures, "AFP_measures.csv")
#write.csv(AFP_measures, "AFP_measures2.csv")
```

# Probability of HCC Equation

P(HCC)i = [1+exp(-c1Bi-c2SDi-c3RRi)^-1]

- Bi: vector of all states cofactors (risk factors that do not vary over time)
- SDi: standard deviation of patient's AFP
- RRi: Least square estimate for the rate of AFP rise over time of a patient
- c1: vector of corresponding regression for the static risk factors
- c2, c3: regression coefficient for AFP standard deviation and rate of rise of AFP over time.

# Calculating Coefficients

## Calculating coefficient c1

```
static_glm <- glm(HCC ~., data = Patients_StaticData2, family = binomial(logit))
c1_coeff <- round(static_glm$coefficients, 2)
c1_coeff
```

```
##         (Intercept)                 Age          Ethnicity2
##               24.81               -0.06              -18.75
##          Ethnicity3          Ethnicity4             Smoker1
##              -17.94               -2.03              -18.65
##  Alkaline_Phosphate     Blood_Platelets Esophaegal_Varices1
##               -0.02                0.00               -1.55
```

## Calculating coefficients c2, c3

I bind HCC values to the first 500 rows of AFP measures (50 patients at times t1 to t10) and regress SD on HCC and RR on HCC.

c2: coefficient of SD (least square estimate)

```
dumi <- AFP_measures[1:500,]
dumi2 <- cbind(dumi, HCC = rep(Patients_StaticData2$HCC, each = 10))
head(dumi2)
```

```
##   Time AFP       SD          RR HCC
## 1    1  95 0.000000  0.00000000   1
## 2    2  88 4.949747 -0.07368421   1
```

```
## 3    3  86 4.725816 -0.09473684   1
## 4    4  93 4.203173 -0.02105263   1
## 5    5  92 3.701351 -0.03157895   1
## 6    6  92 3.346640 -0.03157895   1
```

```
SS_glm <- glm(as.numeric(HCC)~SD, data = dumi2)
c2_coeff <- round(SS_glm$coefficients, 2)
c2_coeff
```

```
## (Intercept)          SD
##        1.76       -0.02
```

c3: coefficient of RR (least square estimate)

```
RR_glm <- glm(as.numeric(HCC)~RR, data = dumi2)
c3_coeff <- round(RR_glm$coefficients, 2)
```

## Pre-Processing Data

I will create a dataframe in which each row represents a patient in time i, (i=1..10). We have 50 patients.

```
idx <- rep(1:50, each=10)
Patient_Static_Dynamic <- Patients_StaticData2[idx,]
Patient_Static_Dynamic <- cbind(Patient_Static_Dynamic, AFP_measures[1:500,])
Patient_Static_Dynamic <- cbind(P_ID = rep(1:50, each=10), Time = Patient_Static_Dynamic[,8], Patient_S
head(Patient_Static_Dynamic, 13)
```

```
##      P_ID Time Age Ethnicity Smoker Alkaline_Phosphate Blood_Platelets
## 69      1    1  57         2      0                176              51
## 69.1    1    2  57         2      0                176              51
## 69.2    1    3  57         2      0                176              51
## 69.3    1    4  57         2      0                176              51
## 69.4    1    5  57         2      0                176              51
## 69.5    1    6  57         2      0                176              51
## 69.6    1    7  57         2      0                176              51
## 69.7    1    8  57         2      0                176              51
## 69.8    1    9  57         2      0                176              51
## 69.9    1   10  57         2      0                176              51
## 76      2    1  64         3      0                138             101
## 76.1    2    2  64         3      0                138             101
## 76.2    2    3  64         3      0                138             101
##      Esophaegal_Varices       SD          RR AFP HCC
## 69                    0 0.000000  0.00000000  95   1
## 69.1                  0 4.949747 -0.07368421  88   1
## 69.2                  0 4.725816 -0.09473684  86   1
## 69.3                  0 4.203173 -0.02105263  93   1
## 69.4                  0 3.701351 -0.03157895  92   1
## 69.5                  0 3.346640 -0.03157895  92   1
## 69.6                  0 3.258688 -0.01052632  94   1
## 69.7                  0 3.136764 -0.06315789  89   1
## 69.8                  0 2.934469 -0.04210526  91   1
## 69.9                  0 4.228212 -0.14736842  81   1
## 76                    0 0.000000  0.00000000  84   1
## 76.1                  0 0.000000  0.00000000  84   1
## 76.2                  0 3.464102  0.07142857  90   1
```

```
#write.csv(Patient_Static_Dynamic, "Patient_Static_Dynamic.csv")
```

Formatting Patient_Static_Dynamic to fit a matrix format

```
Patient_Static_Dynamic <- read.csv("Patient_Static_Dynamic.csv")

Ethinicity1 <- ifelse(Patient_Static_Dynamic$Ethnicity==1, 1, 0)
Ethinicity2 <- ifelse(Patient_Static_Dynamic$Ethnicity==2, 1, 0)
Ethinicity3 <- ifelse(Patient_Static_Dynamic$Ethnicity==3, 1, 0)
Ethinicity4 <- ifelse(Patient_Static_Dynamic$Ethnicity==4, 1, 0)

Smoker0 <- ifelse(Patient_Static_Dynamic$Smoker==0, 1, 0)
Smoker1 <- ifelse(Patient_Static_Dynamic$Smoker==1, 1, 0)

Esophaegal_Varices0 <- ifelse(Patient_Static_Dynamic$Esophaegal_Varices==0, 1, 0)
Esophaegal_Varices1 <- ifelse(Patient_Static_Dynamic$Esophaegal_Varices==1, 1, 0)

Patient_Static_Dynamic_MatVersion <- cbind(Patient_Static_Dynamic[,1:3], Ethinicity1=Ethinicity1, Ethini

#write.csv(Patient_Static_Dynamic_MatVersion, "Patient_Static_Dynamic_MatVersion.csv")
```

## Probability of HCC Equation

### Function P_HCC

```
P_HCC <- function(B, c1, SD, c2, RR, c3){
  (1+exp(B%*%-c1-c2*SD-c3*RR))^-1
}
```

### Vector of coefficient c1

First I write the c1: coefficients (manually) in a 1 by 11 matrix, each row represents the following coefficient: - Age - Ethinicity1 - Ethinicity2 - Ethinicity3 - Ethinicity4 - Smoker0 - Smoker1 - Alkaline_Phosphate - Blood_Platelets - Esophaegal_Varices0 - Esophaegal_Varices1

```
c1 <- matrix(c(c1_coeff["Age"], 0, c1_coeff["Ethnicity2"], c1_coeff["Ethnicity3"], c1_coeff["Ethnicity4
c1
```

```
##         [,1]
##  [1,]  -0.06
##  [2,]   0.00
##  [3,] -18.75
##  [4,] -17.94
##  [5,]  -2.03
##  [6,]   0.00
##  [7,] -18.65
##  [8,]  -0.02
##  [9,]   0.00
## [10,]   0.00
## [11,]  -1.55
```

### 'Coefficients c2, c3

Second I write the c2 and c3

```
c2 <- c2_coeff["SD"]
c2
```

```
##    SD
## -0.02
```

```
c3 <- c3_coeff["RR"]
c3
```

```
##    RR
## -0.06
```

** B (Static) Matrix ** I extract static matrix B:

```
str(Patient_Static_Dynamic_MatVersion)
```

```
## 'data.frame':    500 obs. of  17 variables:
##  $ P_ID               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Time               : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Age                : int  46 46 46 46 46 46 46 46 46 46 ...
##  $ Ethinicity1        : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Ethinicity2        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ethinicity3        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Ethinicity4        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Smoker0            : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Smoker1            : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Alkaline_Phosphate : int  149 149 149 149 149 149 149 149 149 149 ...
##  $ Blood_Platelets    : int  141 141 141 141 141 141 141 141 141 141 ...
##  $ Esophaegal_Varices0: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Esophaegal_Varices1: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ SD                 : num  0 6.36 4.58 5.85 5.41 ...
##  $ RR                 : num  0 -0.0947 -0.0316 -0.1368 -0.0211 ...
##  $ AFP                : int  95 86 92 82 93 88 94 86 86 91 ...
##  $ HCC                : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
B_Static_Matrix <- as.matrix(Patient_Static_Dynamic_MatVersion[,-c(1,2,14:17)])
```

** SD and RR Matrix ** I extract SD matrix and RR matrix:

```
SD_matrix <- as.matrix(Patient_Static_Dynamic_MatVersion[,14])
RR_matrix <- as.matrix(Patient_Static_Dynamic_MatVersion[,15])
```

Writing matrices and output into csv file:

```
#write.csv(B_Static_Matrix, "B_Static_Matrix.csv")
#write.csv(SD_matrix, "SD_matrix.csv")
#write.csv(RR_matrix, "RR_matrix.csv")
```

** Probability of HCC** Calculating Probability of HCC and saving it in csv file:

```
P_HCC_Output <- P_HCC(B_Static_Matrix, c1, SD_matrix, c2, RR_matrix, c3)
#write.csv(P_HCC_Output, "P_HCC_Output.csv")
```

# Probability of HCC Equation Based on Policies

**Policy 1: Myopic Behavior**

**Policy 2: Greedy strategies**

**Policy 3: Interval Estimation strategies**

**Policy 4: Boltzmann exploration**

## Simulation

1. Initial Panel Module
2. Policy Module
3. Imaging Module (This requires simulated data to represent size of tumor s)
4. AFP Reading Module
5. Update Module and/or New Patient Module
6. Patient Exit Module