296B - Full Doc

ZeXing Hu

2023-03-10

```
#setwd("/Users/zexing/Desktop/296B Final/")
#rm(list = ls())
#install.packages("readxl")
library(lubridate)
library(ggplot2)
library(kableExtra)
library(tidyverse)
library(dplyr)
library(readxl, lib.loc = "/Library/Frameworks/R.framework/Versions/4.2/Resources/librar
y")
```

PART 0: NAIC Cleaning

Load the original data and take out unnecessary info

```
naic <- read.csv("/Users/zexing/Desktop/model comparison/NAIC/1000_Path_Subset_1a_Connin
g_GFF_Baseline_Equity_123121.csv", header = TRUE)
naic <- naic[,-c(3:35)] #get rid of spot rates
naic <- naic[,-c(5:24)] #Only keep 1mo,3mo,20year
naic <- naic[,-c(6:15)]
naic <- naic[,-c(24:39)]
naic <- naic[,-c(17:21)]
naic <- naic[,-c(8:11)]
naic <- naic[,-c(12)]
write.csv(naic, file = "naic_cleaning.csv")
```

Further clean the data to make it more concise

```
naic_c <- read.csv("/Users/zexing/Desktop/296B Final/RAW DATA/naic_cleaning.csv",header
= TRUE)
naic_c <- naic_c %>%
  mutate(int_govt_tr = Int.Govt.Bonds.Price + Int.Govt.Bonds.Income) %>%
  mutate(short_govt_tr = Short.Govt.Bonds.Price + Short.Govt.Bonds.Income) %>%
  mutate(lt_govt_tr = Long.Govt.Bonds.Price + Long.Govt.Bonds.Income) %>%
  mutate(large_cap_tr = Large.Cap.Price + Large.Cap.Income)
naic_c <- naic_c[,-c(7:14)]
naic_c <- naic_c %>%
  rename("ust_1m_yield" = 4,
         "ust_3m_yield" = 5,
         "ust_20yr_yield" = 6)
write.csv(naic_c, file = "naic_final.csv")
```

Load our final data

```r
naic <- read.csv("/Users/zexing/Desktop/model comparison/NAIC/naic_final.csv",header = T
RUE)
naic <- naic[,-c(1:2)]
```

Create a list that store all the values correspondent to I_th time

```r
time <- c(1:1200)
path <- c(1:1000)

# Each object contains 1000 values
mylist <- list()
for (i in time) {
  mylist[[i]] <- which(naic$TIME == i)
}
```

Get UST-1m NAIC Simulated Data

```r
df <- data.frame(matrix(ncol = 1200, nrow = 1000))
for (col in time) {
  for (row in path) {
    df[row,col] <- naic$ust_1m_yield[mylist[[col]][[row]]]
  }
}
write.csv(df, file = "naic_ust_1m.csv")
```

```r
# Check we do this correctly
which(naic$TIME == 4)
naic$ust_1m_yield[10814] #10th simulation, time 4
df[10,4] #10th simulation, time 4
# Number match! Did this correctly!
```

Get UST-3m NAIC Simulated Data

```r
df1 <- data.frame(matrix(ncol = 1200, nrow = 1000))
for (col in time) {
  for (row in path) {
    df1[row,col] <- naic$ust_3m_yield[mylist[[col]][[row]]]
  }
}
write.csv(df1, file = "naic_ust_3m.csv")
```

```r
# Check we do this correctly
which(naic$TIME == 4)
naic$ust_3m_yield[10814] #10th simulation, time 4
df1[10,4] #10th simulation, time 4
# Number match! Did this correctly!
```

Get UST-20Yr NAIC Simulated Data

```r
df2 <- data.frame(matrix(ncol = 1200, nrow = 1000))
for (col in time) {
  for (row in path) {
    df2[row,col] <- naic$ust_20yr_yield[mylist[[col]][[row]]]
  }
}
write.csv(df2, file = "naic_ust_20yr.csv")
```

```r
# Check we do this correctly
which(naic$TIME == 4)
naic$ust_20yr_yield[10814] #10th simulation, time 4
df2[10,4] #10th simulation, time 4
# Number match! Did this correctly!
```

Get Intermediate Govt Bond TR NAIC Simulated Data

```r
df3 <- data.frame(matrix(ncol = 1200, nrow = 1000))
for (col in time) {
  for (row in path) {
    df3[row,col] <- naic$int_govt_tr[mylist[[col]][[row]]]
  }
}
write.csv(df3, file = "naic_int_govt_tr.csv")
```

```r
# Check we do this correctly
which(naic$TIME == 4)
naic$int_govt_tr[10814] #10th simulation, time 4
df3[10,4] #10th simulation, time 4
# Number match! Did this correctly!
```

Get Short Term Govt Bond TR NAIC Simulated Data

```r
df4 <- data.frame(matrix(ncol = 1200, nrow = 1000))
for (col in time) {
  for (row in path) {
    df4[row,col] <- naic$short_govt_tr[mylist[[col]][[row]]]
  }
}
write.csv(df4, file = "naic_short_govt_tr.csv")
```

```r
# Check we do this correctly
which(naic$TIME == 4)
naic$short_govt_tr[10814] #10th simulation, time 4
df4[10,4] #10th simulation, time 4
# Number match! Did this correctly!
```

Get Long Term Govt Bond TR NAIC Simulated Data

```
df5 <- data.frame(matrix(ncol = 1200, nrow = 1000))
for (col in time) {
  for (row in path) {
    df5[row,col] <- naic$lt_govt_tr[mylist[[col]][[row]]]
  }
}
write.csv(df5, file = "naic_lt_govt_tr.csv")
```

```
# Check we do this correctly
which(naic$TIME == 4)
naic$lt_govt_tr[10814] #10th simulation, time 4
df5[10,4] #10th simulation, time 4
# Number match! Did this correctly!
```

Get Large Cap TR NAIC Simulated Data

```
df6 <- data.frame(matrix(ncol = 1200, nrow = 1000))
for (col in time) {
  for (row in path) {
    df6[row,col] <- naic$large_cap_tr[mylist[[col]][[row]]]
  }
}
write.csv(df6, file = "naic_large_cap_tr.csv")
```

```
# Check we do this correctly
which(naic$TIME == 4)
naic$large_cap_tr[10814] #10th simulation, time 4
df6[10,4] #10th simulation, time 4
# Number match! Did this correctly!
```

PART I: Treasury Return Weighted Percentile (3 Month and 20 Year)

Treasury Return Weighted Percentile - SBBI - 20Year

```
sbbi_ust_20yr <- read_excel("/Users/zexing/Desktop/296B Final/Historical Data/SBBI Data for Capital Markets (2022_12_31).xlsx", sheet = "ust_20yr") %>%
  rename("obs_date" = 1, "sbbi_ust_20yr" = 2) %>%
  mutate(obs_date = as.Date(obs_date,origin = "1899-12-30"))
```

We have 97 years of historical data, so we set 48.5 years as our half life. $\frac{1}{2} = (1-\alpha)^{T_{now}-T_{past}}$ $T_{now} = 97Years = 1164Month$ $T_{past} = 48.5Years = 582Month$

```
a <- 1-(1/2)^(1/582)
w <- NULL
for (i in 1:1164){
  w[i] <- (1-a)^(1164-i)
}
nw <- NULL
for (j in 1:1164){
  nw[j] <- w[j]/sum(w)
}
```

```
# Cbind the original data with the weight
sbbi_ust_20yr <- sbbi_ust_20yr %>%
  mutate (normalized_weight = nw)
# Sort the data based on original data
sbbi_ust_20yr<- sbbi_ust_20yr[order(sbbi_ust_20yr$sbbi_ust_20yr),]
# Create a new column that calculate the running sum of the weights
sbbi_ust_20yr<- sbbi_ust_20yr %>%
  mutate(percentage = cumsum(normalized_weight))
```

```
# min
per_min <- min(sbbi_ust_20yr$sbbi_ust_20yr)
# 1%
per_1 <- sbbi_ust_20yr$sbbi_ust_20yr[which(sbbi_ust_20yr$percentage >= 0.009 & sbbi_ust_
20yr$percentage <= 0.01000001)]
# 5%
per_5 <- sbbi_ust_20yr$sbbi_ust_20yr[which(sbbi_ust_20yr$percentage >= 0.049 & sbbi_ust_
20yr$percentage <= 0.05000001)]
# 15%
per_15 <- sbbi_ust_20yr$sbbi_ust_20yr[which(sbbi_ust_20yr$percentage >= 0.14999 & sbbi_u
st_20yr$percentage <= 0.1501)]
# 30%
per_30 <- sbbi_ust_20yr$sbbi_ust_20yr[which(sbbi_ust_20yr$percentage >= 0.2998 & sbbi_us
t_20yr$percentage <= 0.3001)]
# 50%
per_50 <- sbbi_ust_20yr$sbbi_ust_20yr[which(sbbi_ust_20yr$percentage >= 0.4988 & sbbi_us
t_20yr$percentage <= 0.5001)]
# 70%
per_70 <- sbbi_ust_20yr$sbbi_ust_20yr[which(sbbi_ust_20yr$percentage >= 0.6988 & sbbi_us
t_20yr$percentage <= 0.7001)]
# 85%
per_85 <- sbbi_ust_20yr$sbbi_ust_20yr[which(sbbi_ust_20yr$percentage >= 0.849 & sbbi_ust
_20yr$percentage <= 0.8501)]
# 95%
per_95 <- sbbi_ust_20yr$sbbi_ust_20yr[which(sbbi_ust_20yr$percentage >= 0.9499 & sbbi_us
t_20yr$percentage <= 0.9501)]
# 99%
per_99 <- sbbi_ust_20yr$sbbi_ust_20yr[which(sbbi_ust_20yr$percentage >= 0.989 & sbbi_ust
_20yr$percentage <= 0.9901)]
# Max
per_max <- max(sbbi_ust_20yr$sbbi_ust_20yr)
```

```
Percentile <- as.data.frame(c("Min", "1%", "5%", "15%", "30%", "50%",
                "70%", "85%", "95%", "99%", "Max"), byrow = FALSE)
sbbi_ust_20yr_pew <- as.data.frame(c(per_min,per_1,per_5,per_15,per_30,per_50,
                                     per_70,per_85,per_95,per_99,per_max), byrow = FALS
E)
sbbi_ust_20yr_pew <-cbind(Percentile,sbbi_ust_20yr_pew) %>%
  rename("Percentile" = 1, "sbbi_ust_20yr_pew" = 2)
sbbi_ust_20yr_pew
```

```
##    Percentile sbbi_ust_20yr_pew
## 1         Min         0.0097074
## 2          1%         0.0116606
## 3          5%         0.0200413
## 4         15%         0.0246105
## 5         30%         0.0302910
## 6         50%         0.0437684
## 7         70%         0.0623334
## 8         85%         0.0809630
## 9         95%         0.1095100
## 10        99%         0.1347720
## 11        Max         0.1482360
```

SBBI Steady State - Nearest 10 Years

```
sbbi_ust_20yr<- sbbi_ust_20yr[order(sbbi_ust_20yr$obs_date),]
length(sbbi_ust_20yr$obs_date)
```

```
## [1] 1164
```

```
sbbi_ust_20yr_steady <- sbbi_ust_20yr[c(1045:1164),]
sbbi_ust_20yr_steady$sbbi_ust_20yr
```

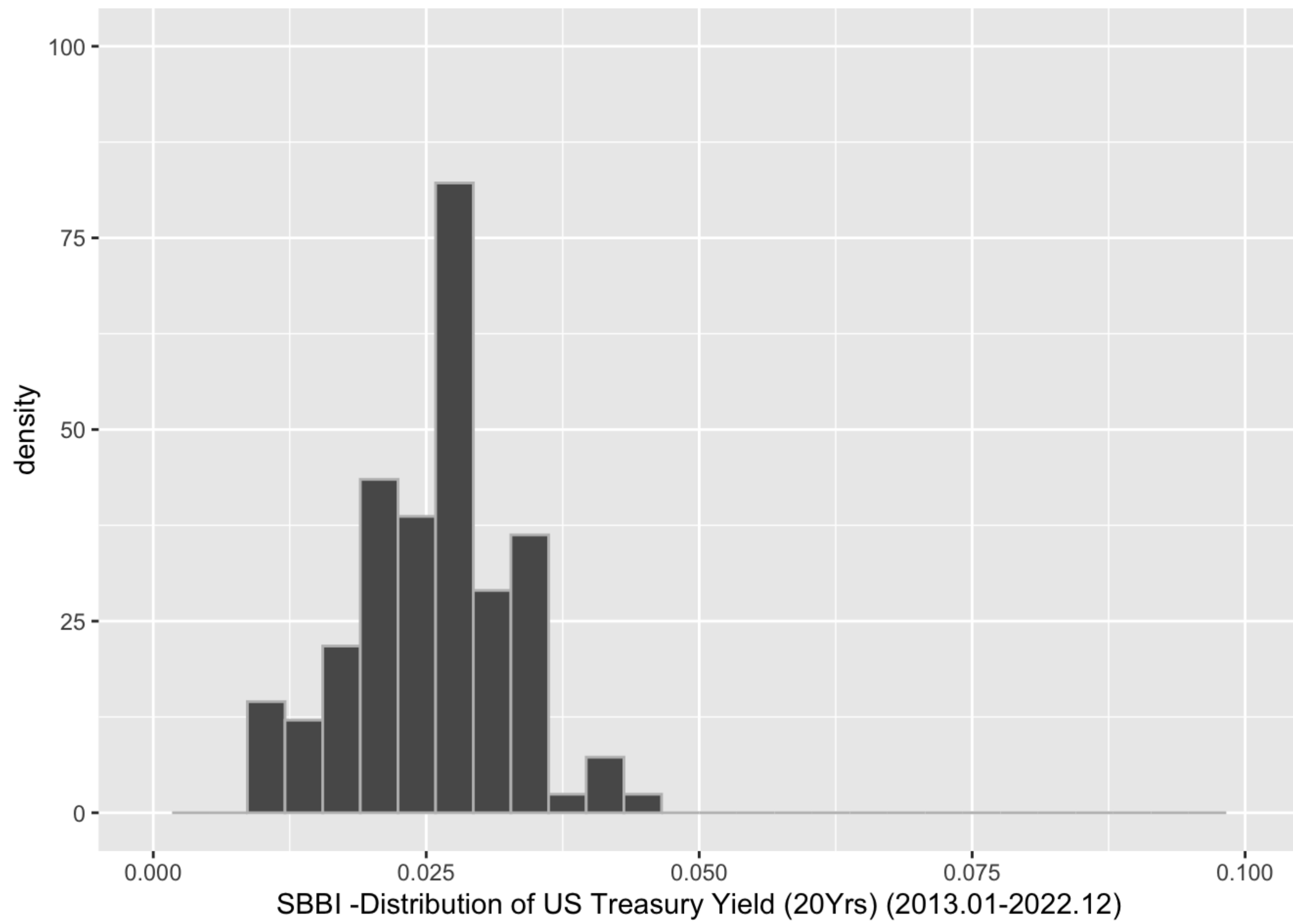
```
##   [1] 0.0291373 0.0285287 0.0286819 0.0264226 0.0308716 0.0329871 0.0343894
##   [8] 0.0351302 0.0349129 0.0342382 0.0360743 0.0377651 0.0342351 0.0339001
##  [15] 0.0336661 0.0326135 0.0309027 0.0312563 0.0310471 0.0286965 0.0300394
##  [22] 0.0281712 0.0263843 0.0245666 0.0200157 0.0238112 0.0230268 0.0249041
##  [29] 0.0261614 0.0284626 0.0263332 0.0264058 0.0253412 0.0258642 0.0264801
##  [36] 0.0267963 0.0236152 0.0216828 0.0218359 0.0223343 0.0218874 0.0179307
##  [43] 0.0174613 0.0185854 0.0195840 0.0220009 0.0266584 0.0272454 0.0278318
##  [50] 0.0269827 0.0274382 0.0265393 0.0255668 0.0257964 0.0261560 0.0242044
##  [57] 0.0258664 0.0261281 0.0260171 0.0254125 0.0283604 0.0301477 0.0284681
##  [64] 0.0299501 0.0289430 0.0290551 0.0302013 0.0292810 0.0333993 0.0351815
##  [71] 0.0316614 0.0283855 0.0285675 0.0293719 0.0264245 0.0274954 0.0239678
##  [78] 0.0233719 0.0233480 0.0180916 0.0195590 0.0200413 0.0205762 0.0225271
##  [85] 0.0190223 0.0153044 0.0113744 0.0108965 0.0116050 0.0115185 0.0097074
##  [92] 0.0121310 0.0116606 0.0133534 0.0128000 0.0136595 0.0159383 0.0201299
##  [99] 0.0225475 0.0210192 0.0211378 0.0193025 0.0173187 0.0176675 0.0195550
## [106] 0.0192991 0.0179364 0.0188233 0.0218903 0.0230861 0.0266742 0.0320741
## [113] 0.0336005 0.0343791 0.0327694 0.0358919 0.0415579 0.0451800 0.0411700
## [120] 0.0424400
```

```
sbbi_ust_20yr_steady%>%
  ggplot(aes(x=sbbi_ust_20yr)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "SBBI -Distribution of US Treasury Yield (20Yrs) (2013.01-2022.12)") +
  xlim(0,0.1) + ylim(0,100)
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## ℹ Please use `after_stat(density)` instead.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



Treasury Return Weighted Percentile - Fred Data - 3m

```
fred_ust_3m <- read_excel("/Users/zexing/Desktop/296B Final/Historical Data/DTB3.xls", s
heet = "ust_3m") %>%
  rename("obs_date" = 1, "fred_ust_3m" = 2) %>%
  mutate(obs_date = as.Date(obs_date,origin = "1899-12-30"))
summary(fred_ust_3m$fred_ust_3m)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.0001  0.0173  0.0400  0.0417  0.0571  0.1552
```

We have 69 years of historical data, but we will use the same half life (48.5years) as the 20 year calculation for consistency.

```
a <- 1-(1/2)^(1/582)
w <- NULL
for (i in 1:829){
  w[i] <- (1-a)^(829-i)
}
nw <- NULL
for (j in 1:829){
  nw[j] <- w[j]/sum(w)
}
```

```
# Cbind the original data with the weight
fred_ust_3m <- fred_ust_3m %>%
  mutate (normalized_weight = nw)
# Sort the data based on original data
fred_ust_3m<- fred_ust_3m[order(fred_ust_3m$fred_ust_3m),]
# Create a new column that calculate the running sum of the weights
fred_ust_3m<- fred_ust_3m %>%
  mutate(percentage = cumsum(normalized_weight))
```

```
# min
per_min <- min(fred_ust_3m$fred_ust_3m)
# 1%
per_1 <- fred_ust_3m$fred_ust_3m[which(fred_ust_3m$percentage >= 0.009 & fred_ust_3m$per
centage <= 0.01000001)]
# 5%
per_5 <- fred_ust_3m$fred_ust_3m[which(fred_ust_3m$percentage >= 0.048 & fred_ust_3m$per
centage <= 0.0501)]
# 15%
per_15 <- fred_ust_3m$fred_ust_3m[which(fred_ust_3m$percentage >= 0.14999 & fred_ust_3m
$percentage <= 0.1501)]
# 30%
per_30 <- fred_ust_3m$fred_ust_3m[which(fred_ust_3m$percentage >= 0.298 & fred_ust_3m$pe
rcentage <= 0.3001)]
# 50%
per_50 <- fred_ust_3m$fred_ust_3m[which(fred_ust_3m$percentage >= 0.4998 & fred_ust_3m$p
ercentage <= 0.5001)]
# 70%
per_70 <- fred_ust_3m$fred_ust_3m[which(fred_ust_3m$percentage >= 0.6988 & fred_ust_3m$p
ercentage <= 0.7001)]
# 85%
per_85 <- fred_ust_3m$fred_ust_3m[which(fred_ust_3m$percentage >= 0.849 & fred_ust_3m$pe
rcentage <= 0.8501)]
# 95%
per_95 <- fred_ust_3m$fred_ust_3m[which(fred_ust_3m$percentage >= 0.949 & fred_ust_3m$pe
rcentage <= 0.9501)]
# 99%
per_99 <- fred_ust_3m$fred_ust_3m[which(fred_ust_3m$percentage >= 0.989 & fred_ust_3m$pe
rcentage <= 0.9901)]
# Max
per_max <- max(fred_ust_3m$fred_ust_3m)
```

```
Percentile <- as.data.frame(c("Min", "1%", "5%", "15%", "30%", "50%",
                "70%", "85%", "95%", "99%", "Max"), byrow = FALSE)
fred_ust_3m_pew <- as.data.frame(c(per_min,per_1,per_5,per_15,per_30,per_50,
                                     per_70,per_85,per_95,per_99,per_max), byrow = FALS
E)
fred_ust_3m_pew <-cbind(Percentile,fred_ust_3m_pew) %>%
  rename("Percentile" = 1, "fred_ust_3m_pew" = 2)
fred_ust_3m_pew
```

```
##    Percentile fred_ust_3m_pew
## 1         Min         -0.0001
## 2          1%          0.0001
## 3          5%          0.0004
## 4         15%          0.0014
## 5         30%          0.0156
## 6         50%          0.0355
## 7         70%          0.0511
## 8         85%          0.0697
## 9         95%          0.0918
## 10        99%          0.1422
## 11        Max          0.1552
```

```
fred_ust_3m<- fred_ust_3m[order(fred_ust_3m$obs_date),]
length(fred_ust_3m$obs_date)
```

```
## [1] 829
```

```
fred_ust_3m_steady <- fred_ust_3m[c(709:828),]
fred_ust_3m_steady$fred_ust_3m
```

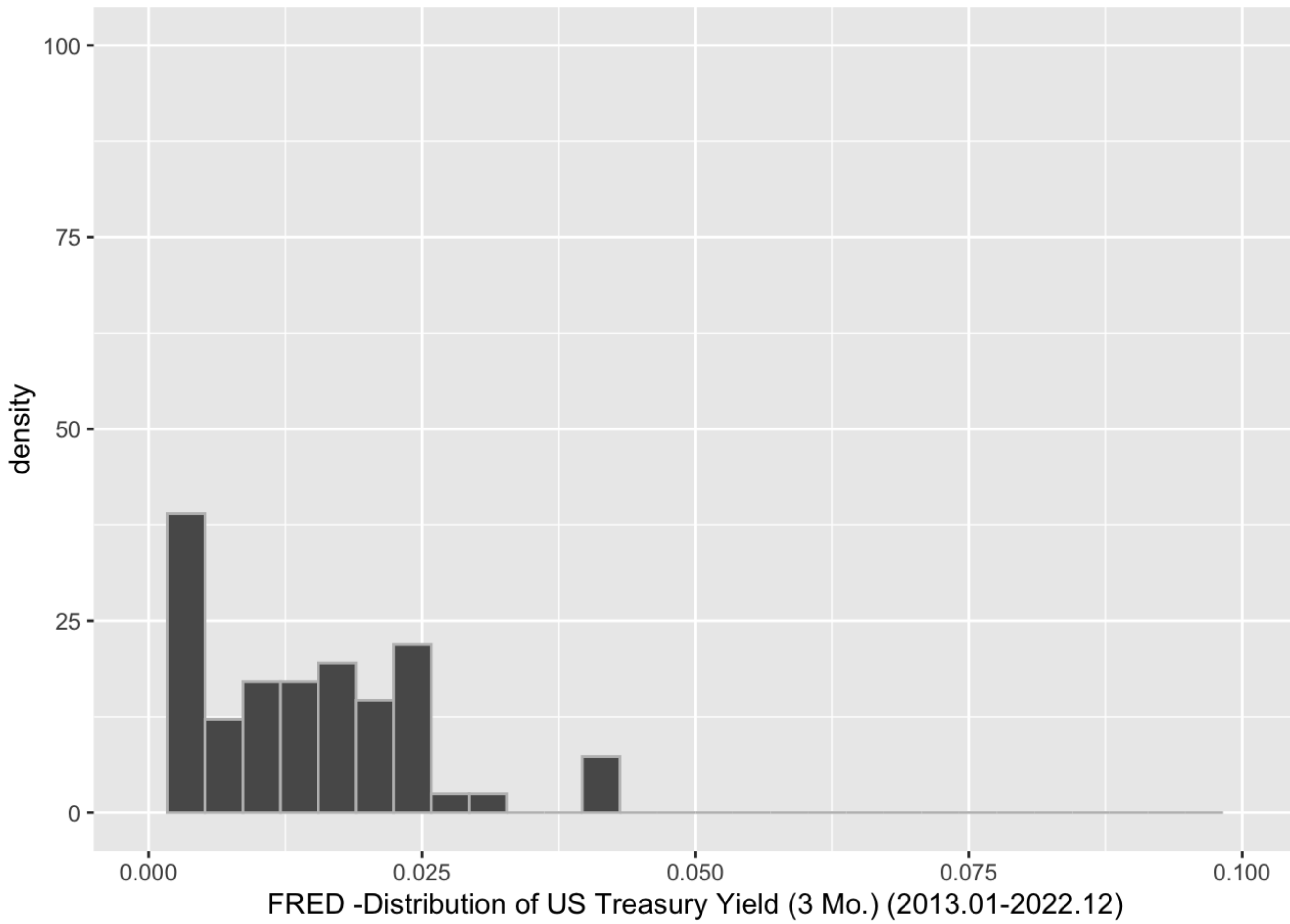
```
##   [1]  0.0007  0.0011  0.0007  0.0005  0.0004  0.0004  0.0004  0.0003  0.0002
##  [10]  0.0004  0.0006  0.0007  0.0002  0.0005  0.0005  0.0003  0.0004  0.0004
##  [19]  0.0003  0.0003  0.0002  0.0001  0.0002  0.0004  0.0002  0.0002  0.0003
##  [28]  0.0001  0.0001  0.0001  0.0008  0.0008 -0.0001  0.0008  0.0022  0.0016
##  [37]  0.0032  0.0033  0.0021  0.0022  0.0034  0.0026  0.0027  0.0033  0.0028
##  [46]  0.0034  0.0048  0.0050  0.0052  0.0053  0.0075  0.0079  0.0096  0.0101
##  [55]  0.0105  0.0099  0.0104  0.0113  0.0125  0.0137  0.0144  0.0163  0.0170
##  [64]  0.0184  0.0189  0.0189  0.0199  0.0207  0.0215  0.0229  0.0232  0.0240
##  [73]  0.0236  0.0240  0.0235  0.0238  0.0230  0.0208  0.0204  0.0195  0.0184
##  [82]  0.0151  0.0156  0.0152  0.0152  0.0125  0.0011  0.0009  0.0014  0.0016
##  [91]  0.0009  0.0011  0.0010  0.0009  0.0008  0.0009  0.0006  0.0004  0.0003
## [100]  0.0001  0.0001  0.0005  0.0006  0.0004  0.0004  0.0005  0.0005  0.0006
## [109]  0.0024  0.0037  0.0051  0.0083  0.0113  0.0166  0.0234  0.0287  0.0322
## [118]  0.0406  0.0427  0.0430
```

```
fred_ust_3m_steady%>%
  ggplot(aes(x=fred_ust_3m)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "FRED -Distribution of US Treasury Yield (3 Mo.) (2013.01-2022.12)")+
  xlim(0,0.1) + ylim(0,100)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



Treasury Return Weighted Percentile - AIRG - 20Yr

```
airg_ust_20yr <- read.csv("/Users/zexing/Desktop/296B Final/AIRG/UST_20y.csv",
                          header = FALSE)
airg_ust_20yr <-  airg_ust_20yr[,-c(1,830:841)] #take out the last year bc I simulate 70
years of data, but realize only need 69
```

Use AIRG data from 1954.01.01 to 2023.01.01. This is 69 years of data. Use the same half life (48.5 years) for consistency.

```
a <- 1-(1/2)^(1/582)
# ncol(airg_ust_20yr) = 828
w <- NULL
for (i in 1:828){
  w[i] <- (1-a)^(828-i)
}
nw <- NULL
for (j in 1:828){
  nw[j] <- w[j]/sum(w)
}
```

```r
wdf <- data.frame(matrix(w, ncol=length(w), byrow=FALSE))
for (row in 1:1000) {
  wdf[row,] <- wdf[1,]
}
nor_wdf <- data.frame(matrix(rep(0), ncol=828, byrow=FALSE))
for (col in 1:828) {
  nor_wdf[,col] <- wdf[1,col]/sum(wdf)
}
for (row in 1:1000) {
  nor_wdf[row,] <- nor_wdf[1,]
}
#sum(nor_wdf) = 1
```

```r
# Condense the orignial data matrix to a vector
airg_ust_20yr <-data.frame(x=unlist(airg_ust_20yr)) %>%
  rename("airg_ust_20yr" = 1)
# Condense the normalized weight matrix to a vector
nw <-data.frame(x=unlist(nor_wdf))
```

```r
# Cbind the original data with the weight
airg_ust_20yr <- airg_ust_20yr %>%
  mutate (normalized_weight = nw[,1])
# Sort the data based on original data
airg_ust_20yr<- airg_ust_20yr[order(airg_ust_20yr$airg_ust_20yr),]
# Create a new column that calculate the running sum of the weights
airg_ust_20yr<- airg_ust_20yr %>%
  mutate(percentage = cumsum(normalized_weight))
```

```
# min
per_min <- min(airg_ust_20yr$airg_ust_20yr)
# 1%
per_1 <- airg_ust_20yr$airg_ust_20yr[which(airg_ust_20yr$percentage >= 0.009998 & airg_u
st_20yr$percentage <= 0.0100)]
# 5%
per_5 <- airg_ust_20yr$airg_ust_20yr[which(airg_ust_20yr$percentage >= 0.0499999 & airg_
ust_20yr$percentage <= 0.0500)]
# 15%
per_15 <- airg_ust_20yr$airg_ust_20yr[which(airg_ust_20yr$percentage >= 0.149999 & airg_
ust_20yr$percentage <= 0.15000)]
# 30%
per_30 <- airg_ust_20yr$airg_ust_20yr[which(airg_ust_20yr$percentage >= 0.299998 & airg_
ust_20yr$percentage <= 0.3000)]
# 50%
per_50 <- airg_ust_20yr$airg_ust_20yr[which(airg_ust_20yr$percentage >= 0.499998 & airg_
ust_20yr$percentage <= 0.5000)]
# 70%
per_70 <- airg_ust_20yr$airg_ust_20yr[which(airg_ust_20yr$percentage >= 0.6999998 & airg
_ust_20yr$percentage <= 0.70000)]
# 85%
per_85 <- airg_ust_20yr$airg_ust_20yr[which(airg_ust_20yr$percentage >= 0.849999 & airg_
ust_20yr$percentage <= 0.850000)]
# 95%
per_95 <- airg_ust_20yr$airg_ust_20yr[which(airg_ust_20yr$percentage >= 0.949999 & airg_
ust_20yr$percentage <= 0.95000)]
# 99%
per_99 <- airg_ust_20yr$airg_ust_20yr[which(airg_ust_20yr$percentage >= 0.9899978 & airg
_ust_20yr$percentage <= 0.99000)]
# Max
per_max <- max(airg_ust_20yr$airg_ust_20yr)
```

```
Percentile <- as.data.frame(c("Min", "1%", "5%", "15%", "30%", "50%",
                "70%", "85%", "95%", "99%", "Max"), byrow = FALSE)
airg_ust_20yr_pew <- as.data.frame(c(per_min,per_1,per_5,per_15,per_30,per_50,
                                     per_70,per_85,per_95,per_99,per_max), byrow = FALS
E)
airg_ust_20yr_pew <-cbind(Percentile,airg_ust_20yr_pew) %>%
  rename("Percentile" = 1, "airg_ust_20yr_pew" = 2)
airg_ust_20yr_pew
```

```
##    Percentile airg_ust_20yr_pew
## 1         Min           0.00917
## 2          1%           0.01441
## 3          5%           0.01812
## 4         15%           0.02176
## 5         30%           0.02538
## 6         50%           0.02946
## 7         70%           0.03451
## 8         85%           0.04102
## 9         95%           0.05156
## 10        99%           0.06881
## 11        Max           0.16860
```

Steady State - AIRG 20 Year

```
airg_ust_20yr <- read.csv("/Users/zexing/Desktop/296B Final/AIRG/UST_20y.csv",
                          header = FALSE)
airg_ust_20yr <-  airg_ust_20yr[,-c(1,830:841)]#take out the last year bc I simulate 70 years of data, but realize only need 69
length(airg_ust_20yr)
```

```
## [1] 828
```

```
airg_ust_20yr_steady <- airg_ust_20yr[,c(709:828)] #(2012.12-2022.12)
length(airg_ust_20yr_steady)
```

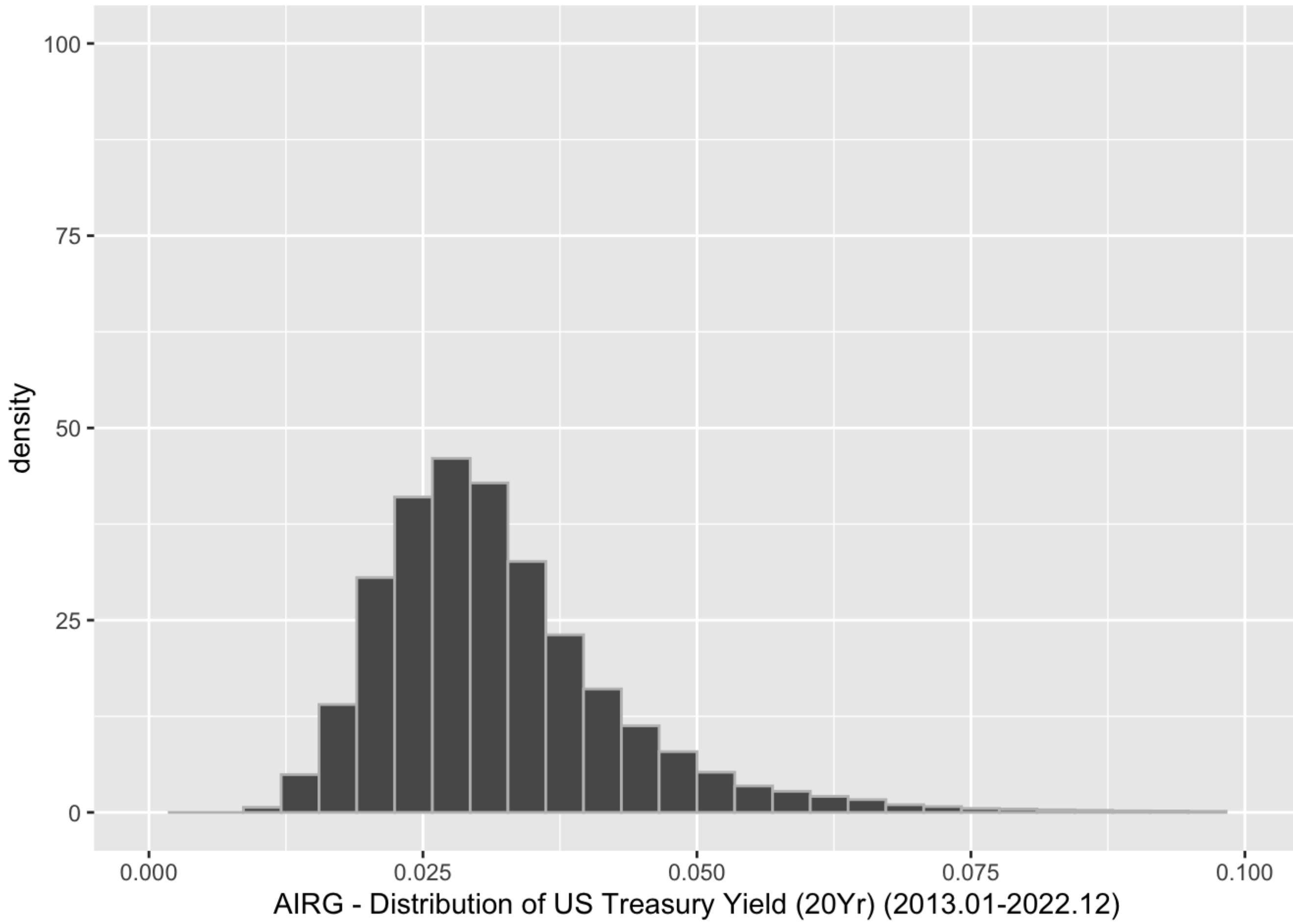
```
## [1] 120
```

```
airg_ust_20yr_steady <- as_tibble(unlist(airg_ust_20yr_steady))
airg_ust_20yr_steady%>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "AIRG - Distribution of US Treasury Yield (20Yr) (2013.01-2022.12)")+
  xlim(0,0.1) + ylim(0,100)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 69 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



Treasury Return Weighted Percentile - AIRG - 3m

```
airg_ust_3m <- read.csv("/Users/zexing/Desktop/296B Final/AIRG/UST_3m.csv",
                          header = FALSE)
airg_ust_3m <-  airg_ust_3m[,-c(1,830:841)] #take out the last year bc I simulate 70 yea
rs of data, but realize only need 69
```

Use AIRG data from 1954.01.01 to 2023.01.01. This is 69 years of data. Use the same half life (48.5 years) for consistency.

```
#eval = FALSE because just reuse the same matrix from the 20 year
a <- 1-(1/2)^(1/582)
# ncol(airg_ust_3m) = 828
w <- NULL
for (i in 1:828){
  w[i] <- (1-a)^(828-i)
}
```

```r
#eval = FALSE because just reuse the same matrix from the 20 year
wdf <- data.frame(matrix(w, ncol=length(w), byrow=FALSE))
for (row in 1:1000) {
  wdf[row,] <- wdf[1,]
}
nor_wdf <- data.frame(matrix(rep(0), ncol=828, byrow=FALSE))
for (col in 1:828) {
  nor_wdf[,col] <- wdf[1,col]/sum(wdf)
}
for (row in 1:1000) {
  nor_wdf[row,] <- nor_wdf[1,]
}
#sum(nor_wdf) = 1
# Condense the normalized weight matrix to a vector
nw <-data.frame(x=unlist(nor_wdf))
```

```r
# Condense the orignial data matrix to a vector
airg_ust_3m <-data.frame(x=unlist(airg_ust_3m)) %>%
  rename("airg_ust_3m" = 1)
```

```r
# Cbind the original data with the weight
airg_ust_3m <- airg_ust_3m %>%
  mutate (normalized_weight = nw[,1])
# Sort the data based on original data
airg_ust_3m<- airg_ust_3m[order(airg_ust_3m$airg_ust_3m),]
# Create a new column that calculate the running sum of the weights
airg_ust_3m<- airg_ust_3m %>%
  mutate(percentage = cumsum(normalized_weight))
```

```
# min
per_min <- min(airg_ust_3m$airg_ust_3m)
# 1%
per_1 <- airg_ust_3m$airg_ust_3m[which(airg_ust_3m$percentage >= 0.009998 & airg_ust_3m
$percentage <= 0.0100)]
# 5%
per_5 <- airg_ust_3m$airg_ust_3m[which(airg_ust_3m$percentage >= 0.049998 & airg_ust_3m
$percentage <= 0.0500)]
# 15%
per_15 <- airg_ust_3m$airg_ust_3m[which(airg_ust_3m$percentage >= 0.149998 & airg_ust_3m
$percentage <= 0.15000)]
# 30%
per_30 <- airg_ust_3m$airg_ust_3m[which(airg_ust_3m$percentage >= 0.299999 & airg_ust_3m
$percentage <= 0.3000)]
# 50%
per_50 <- airg_ust_3m$airg_ust_3m[which(airg_ust_3m$percentage >= 0.499999 & airg_ust_3m
$percentage <= 0.5000)]
# 70%
per_70 <- airg_ust_3m$airg_ust_3m[which(airg_ust_3m$percentage >= 0.699999 & airg_ust_3m
$percentage <= 0.70000)]
# 85%
per_85 <- airg_ust_3m$airg_ust_3m[which(airg_ust_3m$percentage >= 0.849999 & airg_ust_3m
$percentage <= 0.850000)]
# 95%
per_95 <- airg_ust_3m$airg_ust_3m[which(airg_ust_3m$percentage >= 0.949999 & airg_ust_3m
$percentage <= 0.95000)]
# 99%
per_99 <- airg_ust_3m$airg_ust_3m[which(airg_ust_3m$percentage >= 0.989999 & airg_ust_3m
$percentage <= 0.99000)]
# Max
per_max <- max(airg_ust_3m$airg_ust_3m)
```

```
Percentile <- as.data.frame(c("Min", "1%", "5%", "15%", "30%", "50%",
                "70%", "85%", "95%", "99%", "Max"), byrow = FALSE)
airg_ust_3m_pew <- as.data.frame(c(per_min,per_1,per_5,per_15,per_30,per_50,
                                     per_70,per_85,per_95,per_99,per_max), byrow = FALS
E)
airg_ust_3m_pew <-cbind(Percentile,airg_ust_3m_pew) %>%
  rename("Percentile" = 1, "airg_ust_3m_pew" = 2)
airg_ust_3m_pew
```

```
##    Percentile airg_ust_3m_pew
## 1         Min         0.00010
## 2          1%         0.00010
## 3          5%         0.00387
## 4         15%         0.00824
## 5         30%         0.01232
## 6         50%         0.01729
## 7         70%         0.02339
## 8         85%         0.03098
## 9         95%         0.04329
## 10        99%         0.06336
## 11        Max         0.16775
```

```
airg_ust_3m <- read.csv("/Users/zexing/Desktop/296B Final/AIRG/UST_3m.csv",
                          header = FALSE)
airg_ust_3m <-  airg_ust_3m[,-c(1,830:841)]
airg_ust_3m_steady <- airg_ust_3m[,c(709:828)] #(2012.12-2022.12)
length(airg_ust_3m_steady)
```

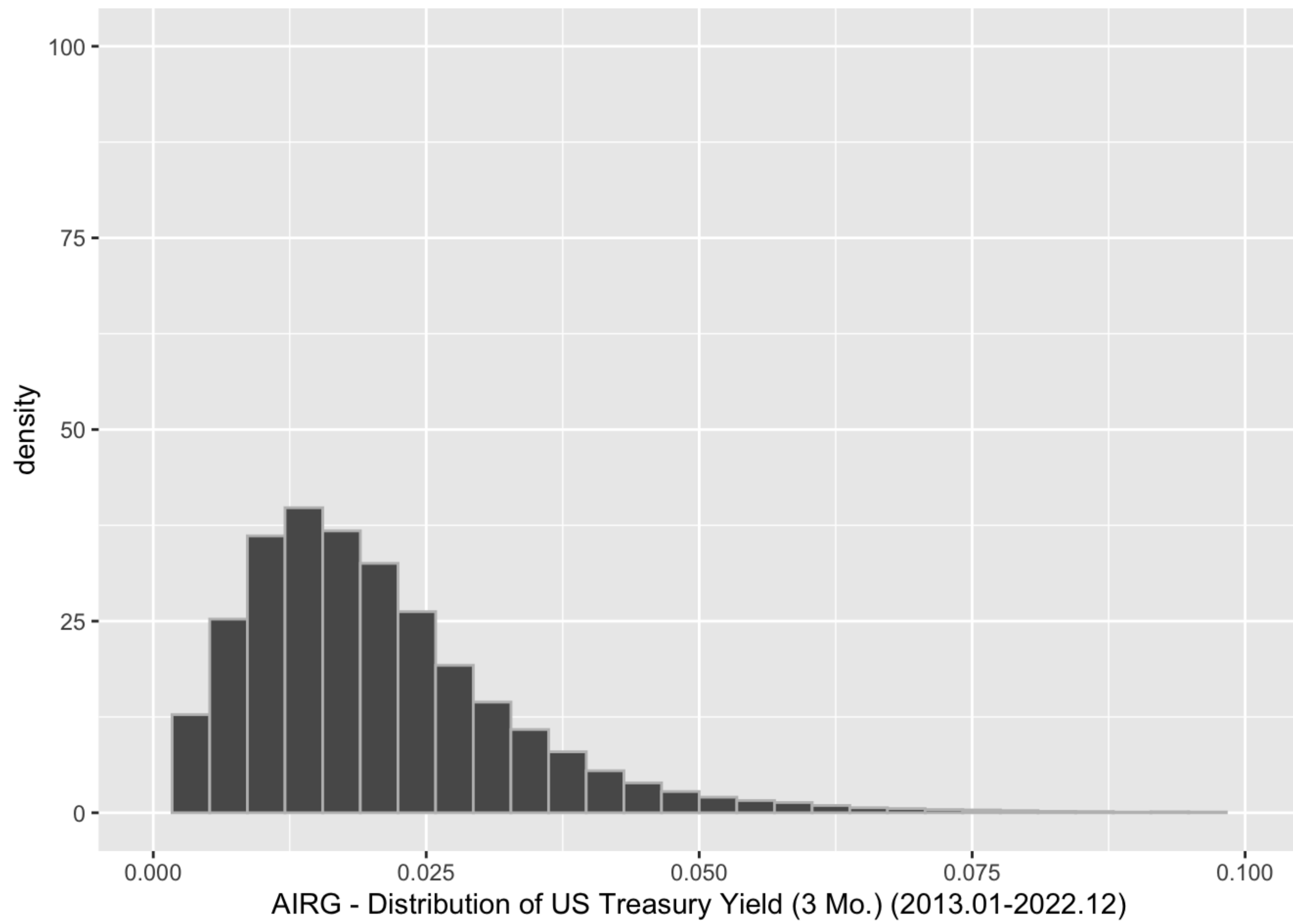
```
## [1] 120
```

```
airg_ust_3m_steady <- as_tibble(unlist(airg_ust_3m_steady))
airg_ust_3m_steady%>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "AIRG - Distribution of US Treasury Yield (3 Mo.) (2013.01-2022.12)")+
  xlim(0,0.1) + ylim(0,100)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 119 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



Treasury Return Weighted Percentile - NAIC - 3m

```
naic_ust_3m <- read.csv("/Users/zexing/Desktop/296B Final/NAIC/naic_ust_3m.csv")
naic_ust_3m <- naic_ust_3m[,-1]
```

NAIC data is given. This contains 100 years of data. Use the same half life (48.5 years) for consistency.

```
a <- 1-(1/2)^(1/582)
# ncol(naic_ust_3m) = 1200
w <- NULL
for (i in 1:ncol(naic_ust_3m)){
  w[i] <- (1-a)^(828-i)
}
```

```r
wdf <- data.frame(matrix(w, ncol=length(w), byrow=FALSE))
for (row in 1:1000) {
  wdf[row,] <- wdf[1,]
}
nor_wdf <- data.frame(matrix(rep(0), ncol=ncol(wdf), byrow=FALSE))
for (col in 1:ncol(wdf)) {
  nor_wdf[,col] <- wdf[1,col]/sum(wdf)
}
for (row in 1:1000) {
  nor_wdf[row,] <- nor_wdf[1,]
}
#sum(nor_wdf) = 1
# Condense the normalized weight matrix to a vector
nw <-data.frame(x=unlist(nor_wdf))
```

```r
# Condense the orignial data matrix to a vector
naic_ust_3m <-data.frame(x=unlist(naic_ust_3m)) %>%
  rename("naic_ust_3m" = 1)
```

```r
# Cbind the original data with the weight
naic_ust_3m <- naic_ust_3m %>%
  mutate (normalized_weight = nw[,1])
# Sort the data based on original data
naic_ust_3m<- naic_ust_3m[order(naic_ust_3m$naic_ust_3m),]
# Create a new column that calculate the running sum of the weights
naic_ust_3m<- naic_ust_3m %>%
  mutate(percentage = cumsum(normalized_weight))
```

```
# min
per_min <- min(naic_ust_3m$naic_ust_3m)
# 1%
per_1 <- naic_ust_3m$naic_ust_3m[which(naic_ust_3m$percentage >= 0.009999 & naic_ust_3m
$percentage <= 0.0100)]
# 5%
per_5 <- naic_ust_3m$naic_ust_3m[which(naic_ust_3m$percentage >= 0.049999 & naic_ust_3m
$percentage <= 0.0500)]
# 15%
per_15 <- naic_ust_3m$naic_ust_3m[which(naic_ust_3m$percentage >= 0.149999 & naic_ust_3m
$percentage <= 0.15000)]
# 30%
per_30 <- naic_ust_3m$naic_ust_3m[which(naic_ust_3m$percentage >= 0.299999 & naic_ust_3m
$percentage <= 0.3000)]
# 50%
per_50 <- naic_ust_3m$naic_ust_3m[which(naic_ust_3m$percentage >= 0.499999 & naic_ust_3m
$percentage <= 0.5000)]
# 70%
per_70 <- naic_ust_3m$naic_ust_3m[which(naic_ust_3m$percentage >= 0.699999 & naic_ust_3m
$percentage <= 0.70000)]
# 85%
per_85 <- naic_ust_3m$naic_ust_3m[which(naic_ust_3m$percentage >= 0.849999 & naic_ust_3m
$percentage <= 0.850000)]
# 95%
per_95 <- naic_ust_3m$naic_ust_3m[which(naic_ust_3m$percentage >= 0.949999 & naic_ust_3m
$percentage <= 0.95000)]
# 99%
per_99 <- naic_ust_3m$naic_ust_3m[which(naic_ust_3m$percentage >= 0.989998 & naic_ust_3m
$percentage <= 0.99000)]
# Max
per_max <- max(naic_ust_3m$naic_ust_3m)
```

```
Percentile <- as.data.frame(c("Min", "1%", "5%", "15%", "30%", "50%",
                "70%", "85%", "95%", "99%", "Max"), byrow = FALSE)
naic_ust_3m_pew <- as.data.frame(c(per_min,per_1,per_5,per_15,per_30,per_50,
                                     per_70,per_85,per_95,per_99,per_max), byrow = FALS
E)
naic_ust_3m_pew <-cbind(Percentile,naic_ust_3m_pew) %>%
  rename("Percentile" = 1, "naic_ust_3m_pew" = 2)
naic_ust_3m_pew
```

```
##    Percentile naic_ust_3m_pew
## 1         Min       -0.011529
## 2          1%       -0.006433
## 3          5%       -0.003541
## 4         15%       -0.000342
## 5         30%        0.002987
## 6         50%        0.019389
## 7         70%        0.043730
## 8         85%        0.071335
## 9         95%        0.110232
## 10        99%        0.163329
## 11        Max        0.380910
```

Steady State - NAIC 3Mo

```
#rm(list = ls())
naic_ust_3m <- read.csv("/Users/zexing/Desktop/296B Final/NAIC/naic_ust_3m.csv")
naic_ust_3m <- naic_ust_3m[,-1]
length(naic_ust_3m)
```

```
## [1] 1200
```

```
naic_ust_3m_steady <- naic_ust_3m[,c(1081:1200)]
length(naic_ust_3m_steady)
```

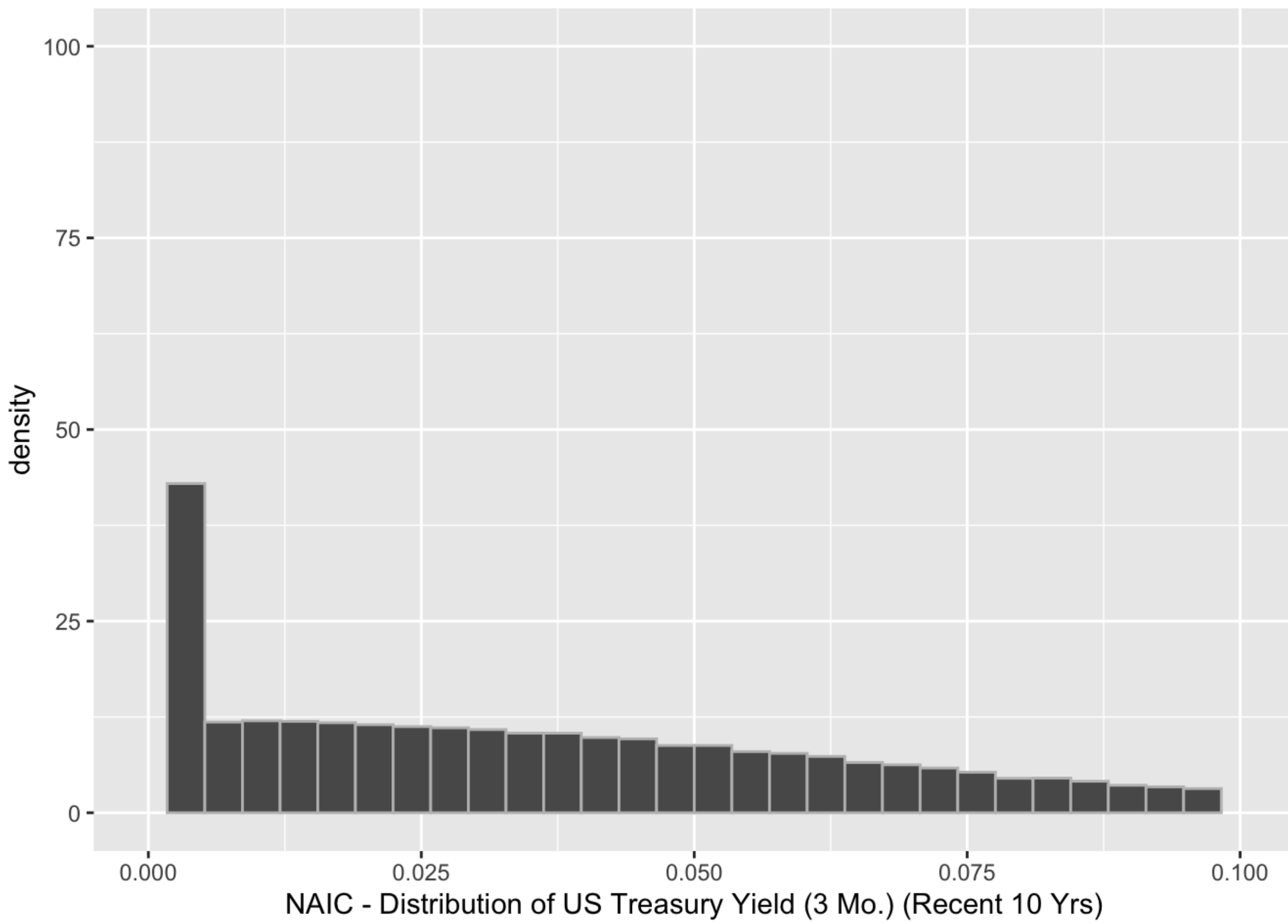
```
## [1] 120
```

```
naic_ust_3m_steady <- as_tibble(unlist(naic_ust_3m_steady))
naic_ust_3m_steady%>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "NAIC - Distribution of US Treasury Yield (3 Mo.) (Recent 10 Yrs)")+
  xlim(0,0.1) + ylim(0,100)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 25957 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



Treasury Return Weighted Percentile - NAIC - 20yr

```
naic_ust_20yr <- read.csv("/Users/zexing/Desktop/296B Final/NAIC/naic_ust_20yr.csv")
naic_ust_20yr <- naic_ust_20yr[,-1]
```

Steady State - NAIC 20Yr

```
#rm(list = ls())
naic_ust_20yr <- read.csv("/Users/zexing/Desktop/296B Final/NAIC/naic_ust_20yr.csv")
naic_ust_20yr <- naic_ust_20yr[,-1]
length(naic_ust_20yr)
```

```
## [1] 1200
```

```
naic_ust_20yr_steady <- naic_ust_20yr[,c(1081:1200)]
length(naic_ust_20yr_steady)
```

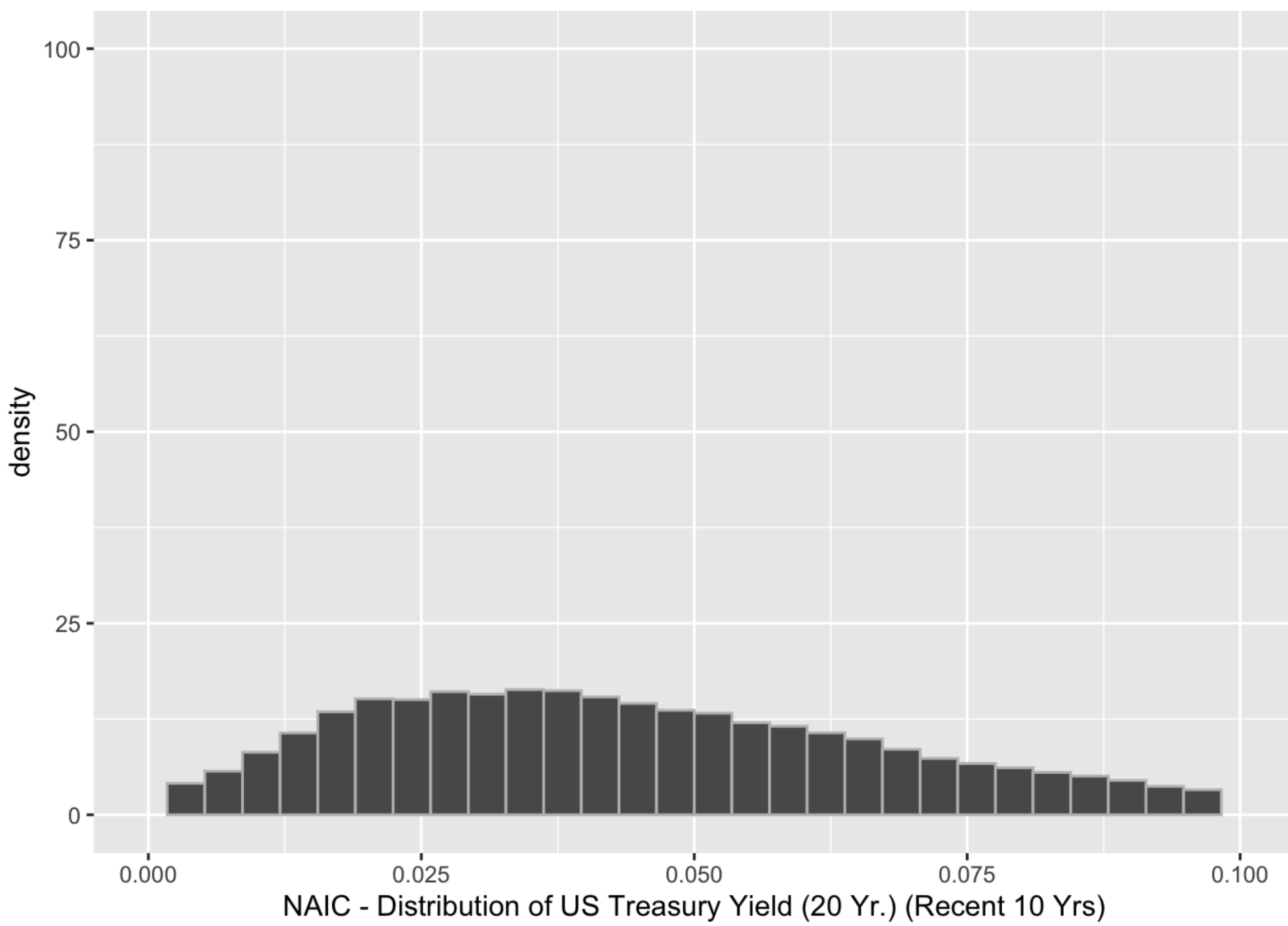
```
## [1] 120
```

```
naic_ust_20yr_steady <- as_tibble(unlist(naic_ust_20yr_steady))
naic_ust_20yr_steady%>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "NAIC - Distribution of US Treasury Yield (20 Yr.) (Recent 10 Yrs)")+
  xlim(0,0.1) + ylim(0,100)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 8300 rows containing non-finite values (`stat_bin()`).
```

```
## Warning: Removed 2 rows containing missing values (`geom_bar()`).
```



NAIC data is given. This contains 100 years of data. Use the same half life (48.5 years) for consistency.

```
# eval = FALSE because will reuse the matrix from 3mo
a <- 1-(1/2)^(1/582)
# ncol(naic_ust_20yr) = 1200
w <- NULL
for (i in 1:ncol(naic_ust_20yr)){
  w[i] <- (1-a)^(828-i)
}
```

```r
# eval = FALSE because will reuse the matrix from 3mo
wdf <- data.frame(matrix(w, ncol=length(w), byrow=FALSE))
for (row in 1:1000) {
  wdf[row,] <- wdf[1,]
}
nor_wdf <- data.frame(matrix(rep(0), ncol=ncol(wdf), byrow=FALSE))
for (col in 1:ncol(wdf)) {
  nor_wdf[,col] <- wdf[1,col]/sum(wdf)
}
for (row in 1:1000) {
  nor_wdf[row,] <- nor_wdf[1,]
}
#sum(nor_wdf) = 1
# Condense the normalized weight matrix to a vector
nw <-data.frame(x=unlist(nor_wdf))
```

```r
# Condense the orignial data matrix to a vector
naic_ust_20yr <-data.frame(x=unlist(naic_ust_20yr)) %>%
  rename("naic_ust_20yr" = 1)
```

```r
# Cbind the original data with the weight
naic_ust_20yr <- naic_ust_20yr %>%
  mutate (normalized_weight = nw[,1])
# Sort the data based on original data
naic_ust_20yr<- naic_ust_20yr[order(naic_ust_20yr$naic_ust_20yr),]
# Create a new column that calculate the running sum of the weights
naic_ust_20yr<- naic_ust_20yr %>%
  mutate(percentage = cumsum(normalized_weight))
```

```
# min
per_min <- min(naic_ust_20yr$naic_ust_20yr)
# 1%
per_1 <- naic_ust_20yr$naic_ust_20yr[which(naic_ust_20yr$percentage >= 0.0099999 & naic_
ust_20yr$percentage <= 0.0100)]
# 5%
per_5 <- naic_ust_20yr$naic_ust_20yr[which(naic_ust_20yr$percentage >= 0.049999 & naic_u
st_20yr$percentage <= 0.0500)]
# 15%
per_15 <- naic_ust_20yr$naic_ust_20yr[which(naic_ust_20yr$percentage >= 0.149999 & naic_
ust_20yr$percentage <= 0.15000)]
# 30%
per_30 <- naic_ust_20yr$naic_ust_20yr[which(naic_ust_20yr$percentage >= 0.299999 & naic_
ust_20yr$percentage <= 0.3000)]
# 50%
per_50 <- naic_ust_20yr$naic_ust_20yr[which(naic_ust_20yr$percentage >= 0.499999 & naic_
ust_20yr$percentage <= 0.5000)]
# 70%
per_70 <- naic_ust_20yr$naic_ust_20yr[which(naic_ust_20yr$percentage >= 0.699999 & naic_
ust_20yr$percentage <= 0.70000)]
# 85%
per_85 <- naic_ust_20yr$naic_ust_20yr[which(naic_ust_20yr$percentage >= 0.849999 & naic_
ust_20yr$percentage <= 0.850000)]
# 95%
per_95 <- naic_ust_20yr$naic_ust_20yr[which(naic_ust_20yr$percentage >= 0.9499998 & naic
_ust_20yr$percentage <= 0.95000)]
# 99%
per_99 <- naic_ust_20yr$naic_ust_20yr[which(naic_ust_20yr$percentage >= 0.9899994 & naic
_ust_20yr$percentage <= 0.99000)]
# Max
per_max <- max(naic_ust_20yr$naic_ust_20yr)
```

```
Percentile <- as.data.frame(c("Min", "1%", "5%", "15%", "30%", "50%",
                "70%", "85%", "95%", "99%", "Max"), byrow = FALSE)
naic_ust_20yr_pew <- as.data.frame(c(per_min,per_1,per_5,per_15,per_30,per_50,
                                     per_70,per_85,per_95,per_99,per_max), byrow = FALS
E)
naic_ust_20yr_pew <-cbind(Percentile,naic_ust_20yr_pew) %>%
  rename("Percentile" = 1, "naic_ust_20yr_pew" = 2)
naic_ust_20yr_pew
```

```
##    Percentile naic_ust_20yr_pew
## 1         Min          0.001422
## 2          1%          0.003820
## 3          5%          0.010031
## 4         15%          0.018068
## 5         30%          0.027203
## 6         50%          0.039868
## 7         70%          0.055963
## 8         85%          0.075472
## 9         95%          0.104528
## 10        99%          0.147869
## 11        Max          0.335166
```

Part II: Equity Risk Premium based on UST-1M

$$Equity\ Risk\ Premium = \frac{Large\ Cap\ TR}{US\ Treasury\ TR_{1mo}} - 1$$

SBBI - 1926.1.1 to 2022.12.01

```
sbbi_large_cap <- read_excel("/Users/zexing/Desktop/296B Final/Historical Data/SBBI Data
for Capital Markets (2022_12_31).xlsx", sheet = "large_cap") %>%
  rename("obs_date" = 1, "sbbi_large_cap" = 2,
         "ust_1m" = 3, "sbbi_equity_return" = 4, "cpi" = 5) %>%
  mutate(obs_date = as.Date(obs_date,origin = "1899-12-30"))
```

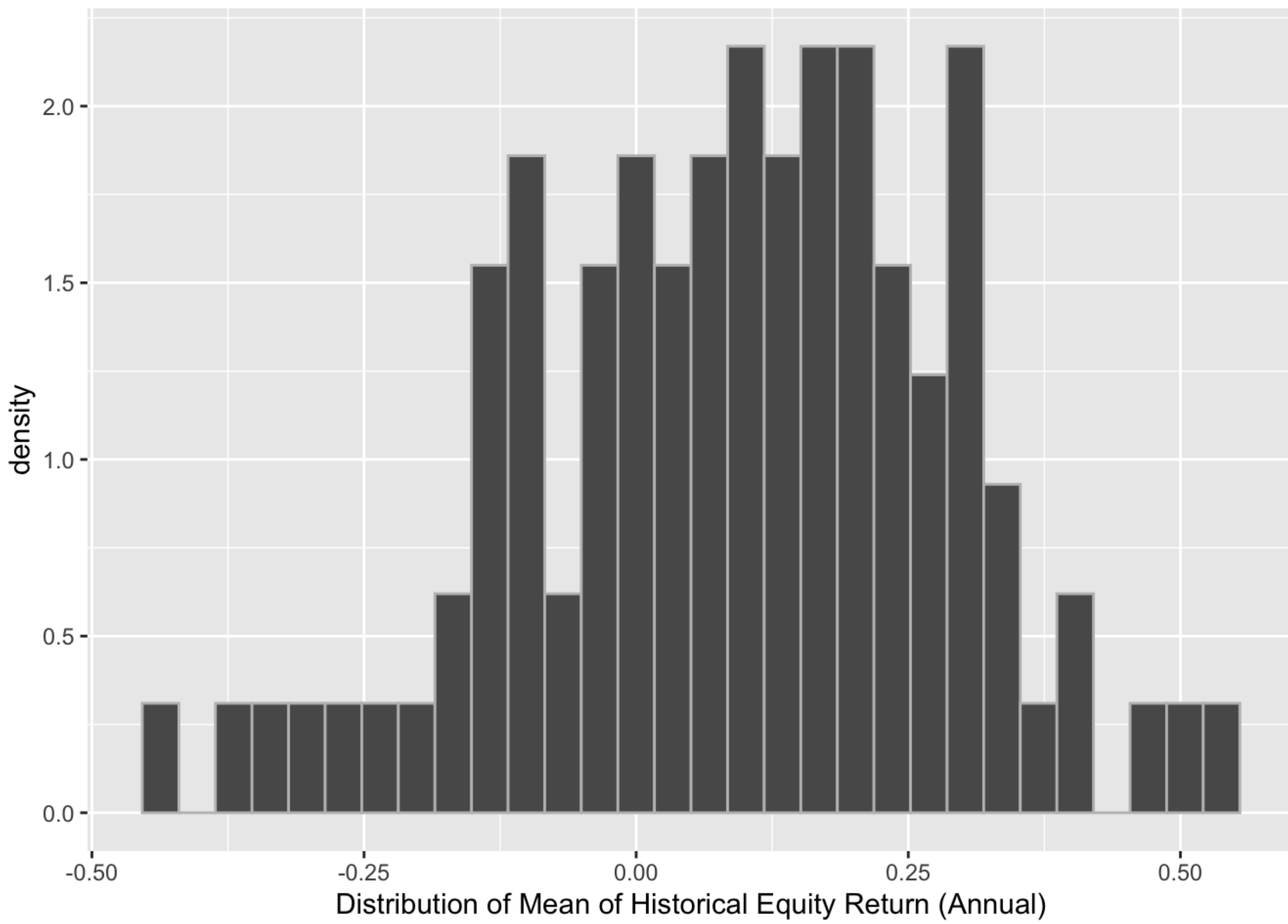
Summary of the overall distribution

```
summary(sbbi_large_cap$sbbi_equity_return)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.43938 -0.02569  0.09801  0.08849  0.22215  0.53533
```

```
sbbi_large_cap %>%
  ggplot(aes(x=sbbi_equity_return))+
  geom_histogram(aes(y=..density..),color = "grey")+
  labs(x = "Distribution of Mean of Historical Equity Return (Annual)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Bucket based on inflation

```
sbbi_large_cap <- sbbi_large_cap[order(sbbi_large_cap$cpi),]
sum(sbbi_large_cap$cpi <=0)
sum((sbbi_large_cap$cpi >0) & (sbbi_large_cap$cpi<=0.025))
sum((sbbi_large_cap$cpi >0.025) & (sbbi_large_cap$cpi<=0.05))
sum(sbbi_large_cap$cpi >=0.05)
sbbi_large_cap <- sbbi_large_cap %>%
  mutate(bucket = c(rep("Deflation",10),
                    rep("Low",35),
                    rep("Medium",33),
                    rep("High",18)))
#sbbi_large_cap <- sbbi_large_cap[order(sbbi_large_cap$obs_date),]
```

Mean of Equity Return within Inflation Bucket

```
sbbi_er_mean <- sbbi_large_cap %>%
    group_by(bucket) %>%
    summarise(sbbi_er_mean = mean(sbbi_equity_return))
sbbi_er_mean <- sbbi_er_mean[c(1,3,4,2),]
sbbi_er_mean
```

```
## # A tibble: 4 × 2
##   bucket    sbbi_er_mean
##   <chr>            <dbl>
## 1 Deflation       0.0999
## 2 Low             0.122
## 3 Medium          0.0878
## 4 High            0.0190
```

NAIC - 100 Years of Data(Given)

First load all the data

```
naic_large_cap <- read.csv("/Users/zexing/Desktop/296B Final/NAIC/naic_large_cap_tr.cs
v")
naic_large_cap <- naic_large_cap[,-1]

naic_ust_1m <-read.csv("/Users/zexing/Desktop/296B Final/NAIC/naic_ust_1m.csv")
naic_ust_1m <- naic_ust_1m[,-1]
summary(unlist(naic_ust_1m))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.012612  0.001149  0.015155  0.029417  0.047157  0.395197
```

Convert to Annual Data

```
naic_large_cap_annual <- data.frame(matrix(1000, ncol=1, nrow = 1000))
for (i in 1:1200){
  naic_large_cap_annual[,i+1] <- naic_large_cap_annual[,i]*(1+naic_large_cap[,i])
}
holder <- data.frame(matrix(0, ncol=100, nrow = 1000))
for (i in 1:100){
  holder[,i] <- naic_large_cap_annual[,1+12*i]/naic_large_cap_annual[,(1+12*(i-1))]-1
}
naic_large_cap_annual <- holder
```

```
summary(sbbi_large_cap$sbbi_large_cap)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.433365 -0.005554  0.146819  0.123297  0.265705  0.539901
```

```
summary(unlist(naic_large_cap_annual))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.607704 -0.002321  0.091312  0.099235  0.191210  1.570281
```

```
naic_ust_1m_annual <- data.frame(matrix(1000, ncol=1, nrow = 1000))
for (i in 1:1200){
  naic_ust_1m_annual[,i+1] <- naic_ust_1m_annual[,i]*(1+naic_ust_1m[,i]/12)
}
holder <- data.frame(matrix(0, ncol=100, nrow = 1000))
for (i in 1:100){
  holder[,i] <- naic_ust_1m_annual[,1+12*i]/naic_ust_1m_annual[,(1+12*(i-1))]-1
}
naic_ust_1m_annual <- holder
```

```
summary(sbbi_large_cap$ust_1m)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0001622  0.0049023  0.0281529  0.0330392  0.0514161  0.1470886
```

```
summary(unlist(naic_ust_1m_annual))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.011422  0.001234  0.015443  0.030501  0.047900  0.430048
```

Then calculate the Equity Return. The formula is (1+large_cap)/(1+ust_1m) - 1

```
naic_er_annual <- data.frame(matrix(0, ncol=100, nrow = 1000))
for (col in 1:100){
  for (row in 1:1000){
    naic_er_annual[row,col] = (1+naic_large_cap_annual[row,col])/(1+naic_ust_1m_annual[r
ow,col]) - 1
  }
}
```

```
summary(sbbi_large_cap$sbbi_equity_return)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.43938 -0.02569  0.09801  0.08849  0.22215  0.53533
```

```
summary(unlist(naic_er_annual))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.62303 -0.02736  0.06146  0.06675  0.15232  1.56076
```

```
x <- unlist(naic_er_annual)
quantile(x,0.01)
```

```
##         1%
## -0.2918397
```

```
quantile(x,0.05)
```

```
##         5%
## -0.1775884
```

```
quantile(x,0.1)
```

```
##        10%
## -0.1182025
```

```
quantile(x,0.9)
```

```
##       90%
## 0.2535841
```

```
quantile(x,0.95)
```

```
##      95%
## 0.327642
```

```
quantile(x,0.99)
```

```
##       99%
## 0.5071646
```

```
quantile(sbbi_large_cap$sbbi_equity_return,0.01)
```

```
##        1%
## -0.382869
```

```
quantile(sbbi_large_cap$sbbi_equity_return,0.05)
```

```
##         5%
## -0.2418802
```

```
quantile(sbbi_large_cap$sbbi_equity_return,0.1)
```

```
##        10%
## -0.1415965
```

```
quantile(sbbi_large_cap$sbbi_equity_return,0.9)
```

```
##       90%
## 0.3071526
```

```
quantile(sbbi_large_cap$sbbi_equity_return,0.95)
```

```
##       95%
## 0.3665956
```

```
quantile(sbbi_large_cap$sbbi_equity_return,0.99)
```

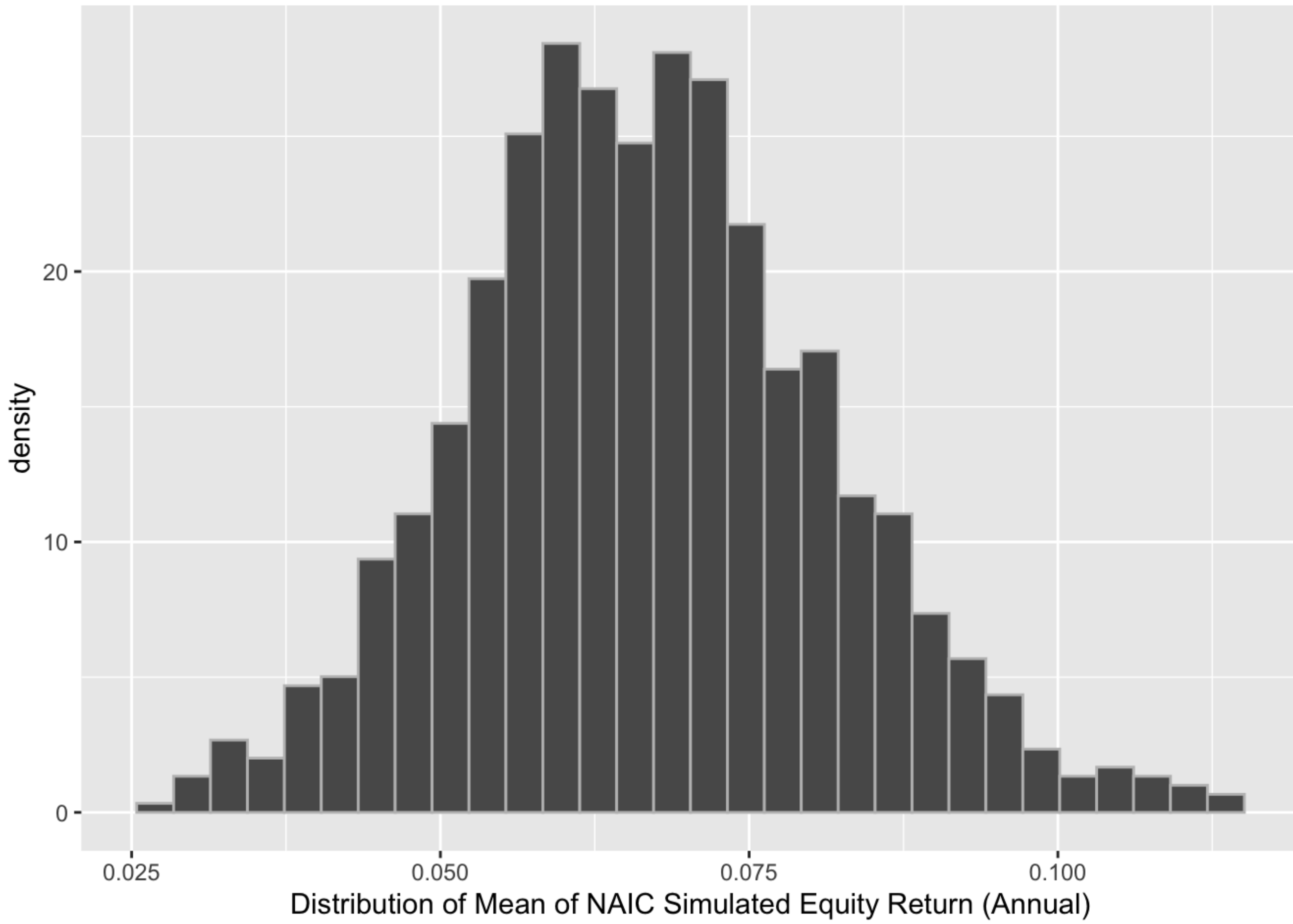
```
##       99%
## 0.5142727
```

```
mean_container <- numeric(1000)
for (i in 1:1000){
  mean_container[i] = mean(t(as.tibble(naic_er_annual[i,])))
}
```

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## ℹ Please use `as_tibble()` instead.
## ℹ The signature and semantics have changed, see `?as_tibble`.
```

```
mean_container <- as.tibble(mean_container)
mean_container %>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
    labs(x = "Distribution of Mean of NAIC Simulated Equity Return (Annual)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



AIRG - 69 Years of Data 1954.1 - 2022.12

AIRG Large Cap (Annual)

```
airg_large_cap <- read.csv("/Users/zexing/Desktop/296B Final/AIRG/airg_large_cap_annual.
csv",header = FALSE)
holder <- data.frame(matrix(0, ncol=70, nrow = 1000))
for (col in 1:70){
  holder[,col] = airg_large_cap[,col+1]/airg_large_cap[,col]-1
}
airg_large_cap <- holder[,-70]#take out the last year bc I simulate 70 years of data, bu
t realize only need 69
```

UST - 3M (Annual)

```
airg_ust_3m <- read.csv("/Users/zexing/Desktop/296B Final/AIRG/airg_ust_3m_annual.csv",
                          header = FALSE)
airg_ust_3m <-  airg_ust_3m[,-c(1,71)]#take out the last year bc I simulate 70 years of
data, but realize only need 69
summary(unlist(airg_ust_3m))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00010 0.01097 0.01714 0.01944 0.02508 0.15993
```

Then calculate the Equity Return. The formula is (1+large_cap)/(1+ust_3m) - 1

```
airg_er_annual <- data.frame(matrix(0, ncol=69, nrow = 1000))
for (col in 1:69){
  for (row in 1:1000){
    airg_er_annual[row,col] = (1+airg_large_cap[row,col])/(1+airg_ust_3m[row,col]) - 1
  }
}
summary(unlist(airg_er_annual))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.61334 -0.03557  0.06376  0.06725  0.16474  1.59825
```

```
x <- unlist(airg_er_annual)
quantile(x,0.01)
```

```
##         1%
## -0.2975915
```

```
quantile(x,0.05)
```

```
##         5%
## -0.1857869
```

```
quantile(x,0.1)
```

```
##        10%
## -0.1279921
```

```
quantile(x,0.9)
```

```
##       90%
## 0.2640601
```

```
quantile(x,0.95)
```

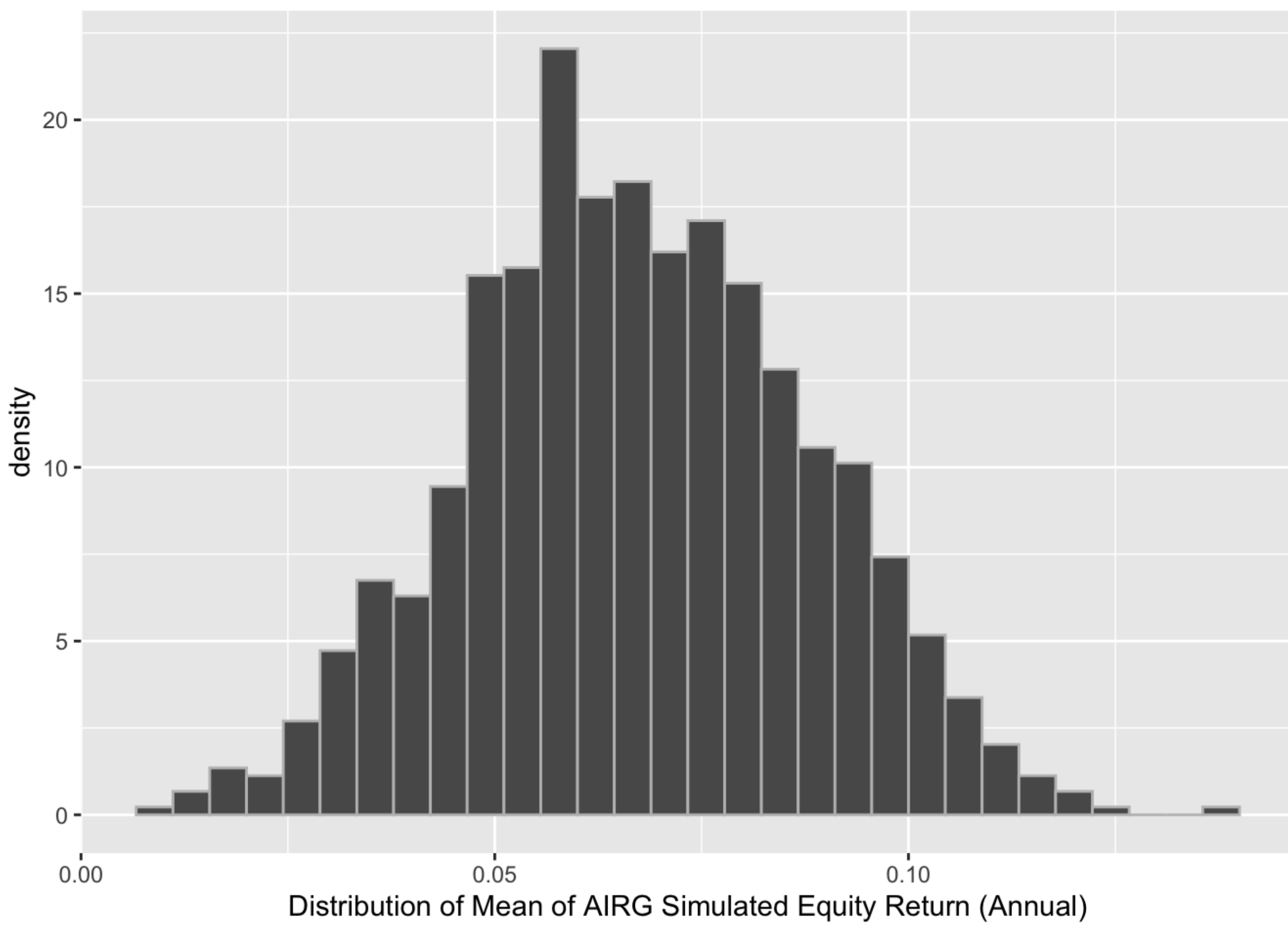
```
##      95%
## 0.331132
```

```
quantile(x,0.99)
```

```
##       99%
## 0.4776289
```

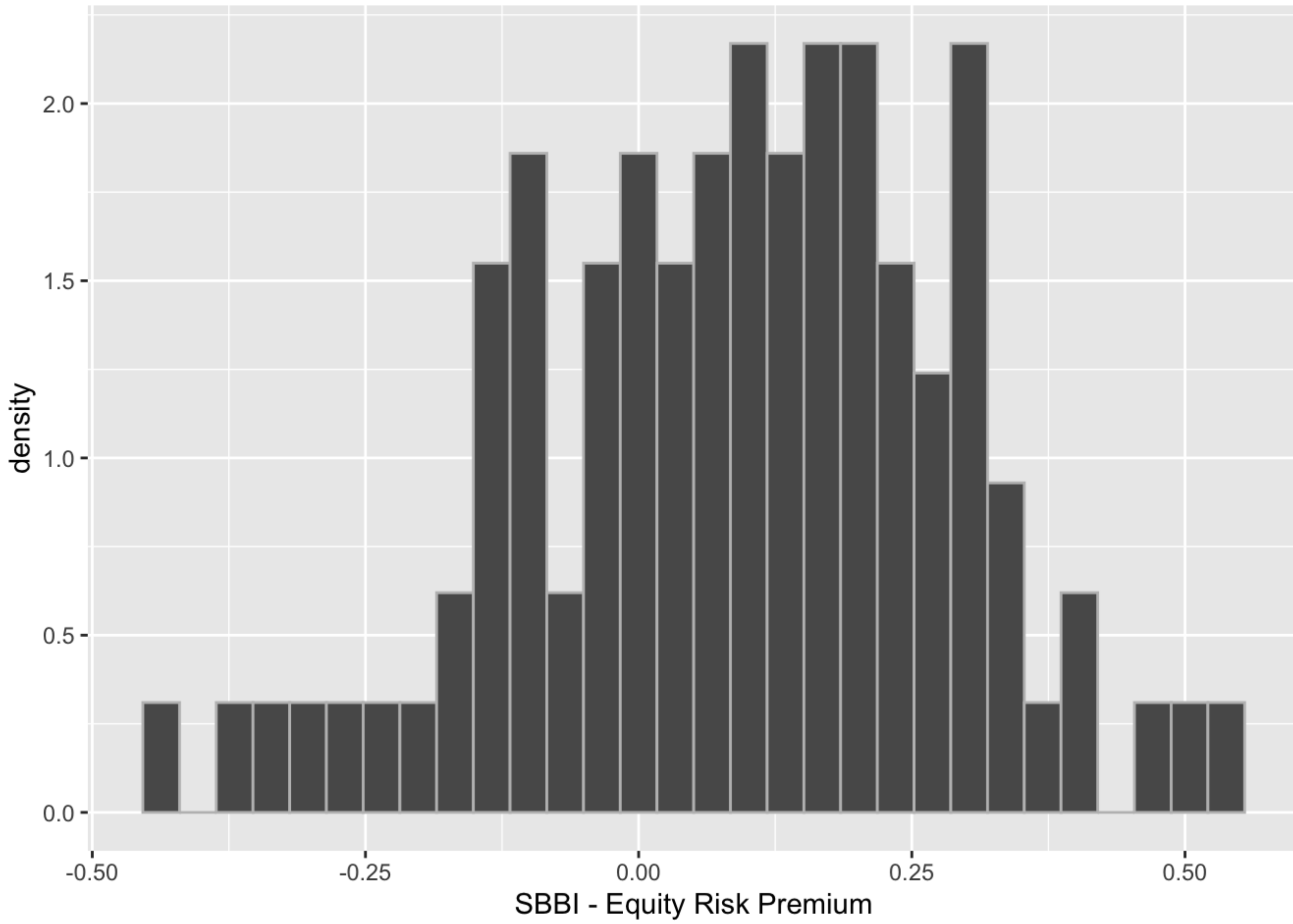
```
mean_container <- numeric(1000)
for (i in 1:1000){
  mean_container[i] = mean(t(as.tibble(airg_er_annual[i,])))
}
mean_container <- as.tibble(mean_container)
mean_container %>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
    labs(x = "Distribution of Mean of AIRG Simulated Equity Return (Annual)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



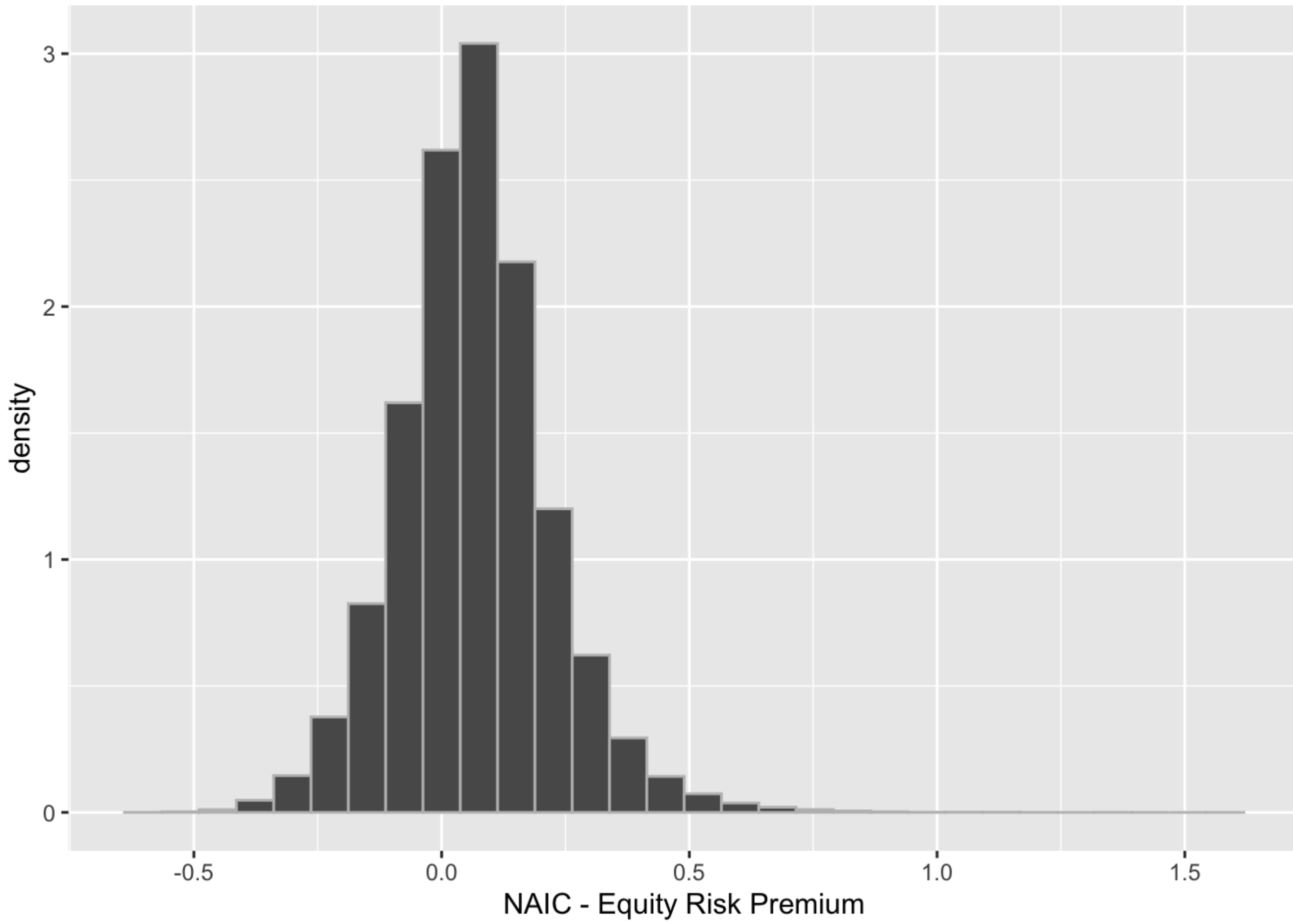

```
sbbi_large_cap %>%
  ggplot(aes(x=sbbi_equity_return)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "SBBI - Equity Risk Premium")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



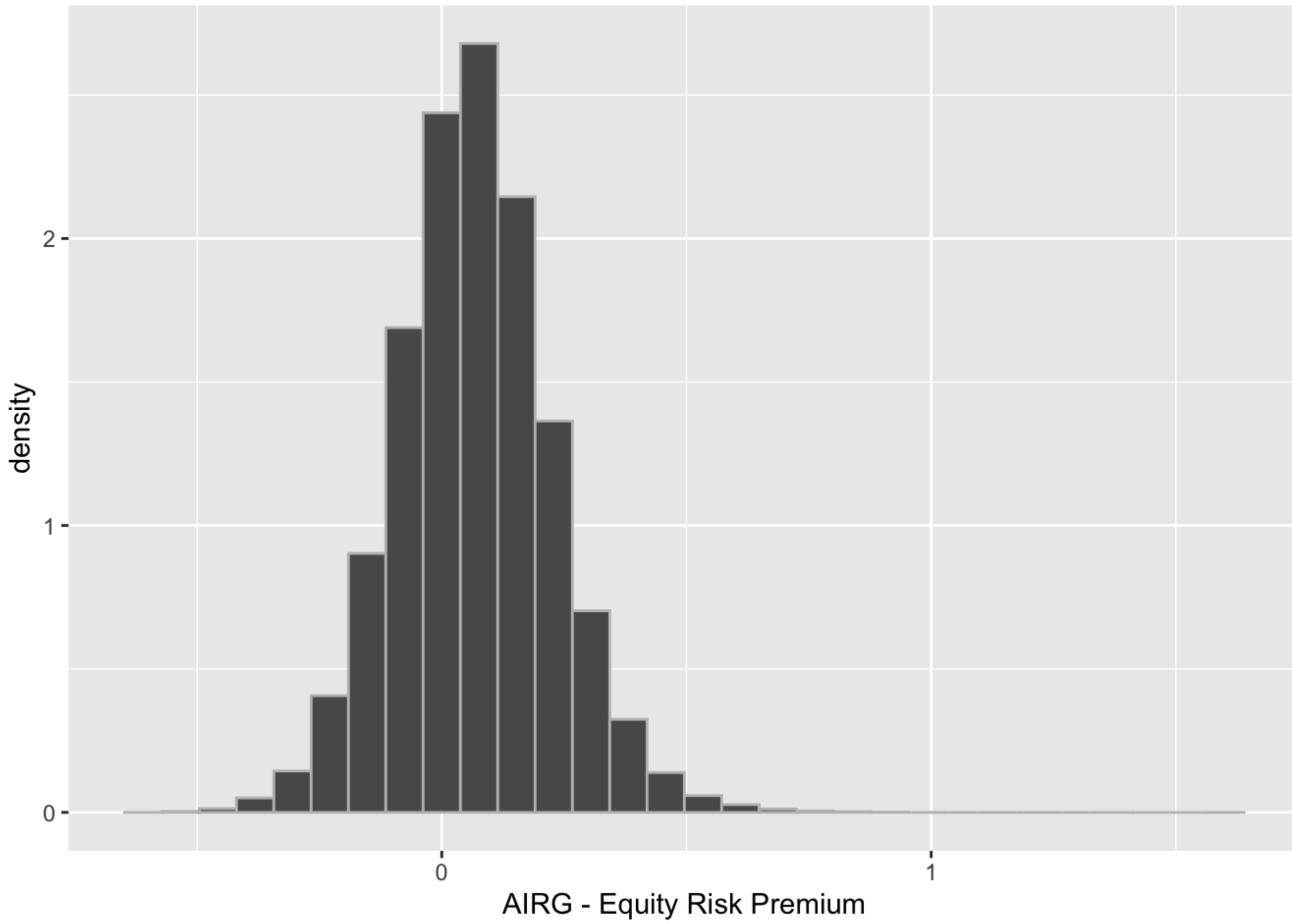

```
naic_er_annual_v <- as.tibble(unlist(naic_er_annual))
naic_er_annual_v%>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "NAIC - Equity Risk Premium")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
airg_er_annual_v <- as.tibble(unlist(airg_er_annual))
airg_er_annual_v %>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "AIRG - Equity Risk Premium")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Part III: Bond Premium based on UST-1M

$$Bond\ Risk\ Premium = \frac{Bond\ Return}{US\ Treasury\ TR_{1mo.}} - 1$$

SBBI - 1926.1.1 to 2022.12.01

SBBI - Intermediate Government Bond (5Yr)

```
sbbi_it_bond <- read_excel("/Users/zexing/Desktop/296B Final/Historical Data/SBBI Data f
or Capital Markets (2022_12_31).xlsx", sheet = "it_bond") %>%
  rename("obs_date" = 1, "sbbi_it_bond" = 2,
         "ust_1m" = 3, "sbbi_it_bond_risk_premium" = 4, "cpi" = 5) %>%
  mutate(obs_date = as.Date(obs_date,origin = "1899-12-30"))
```

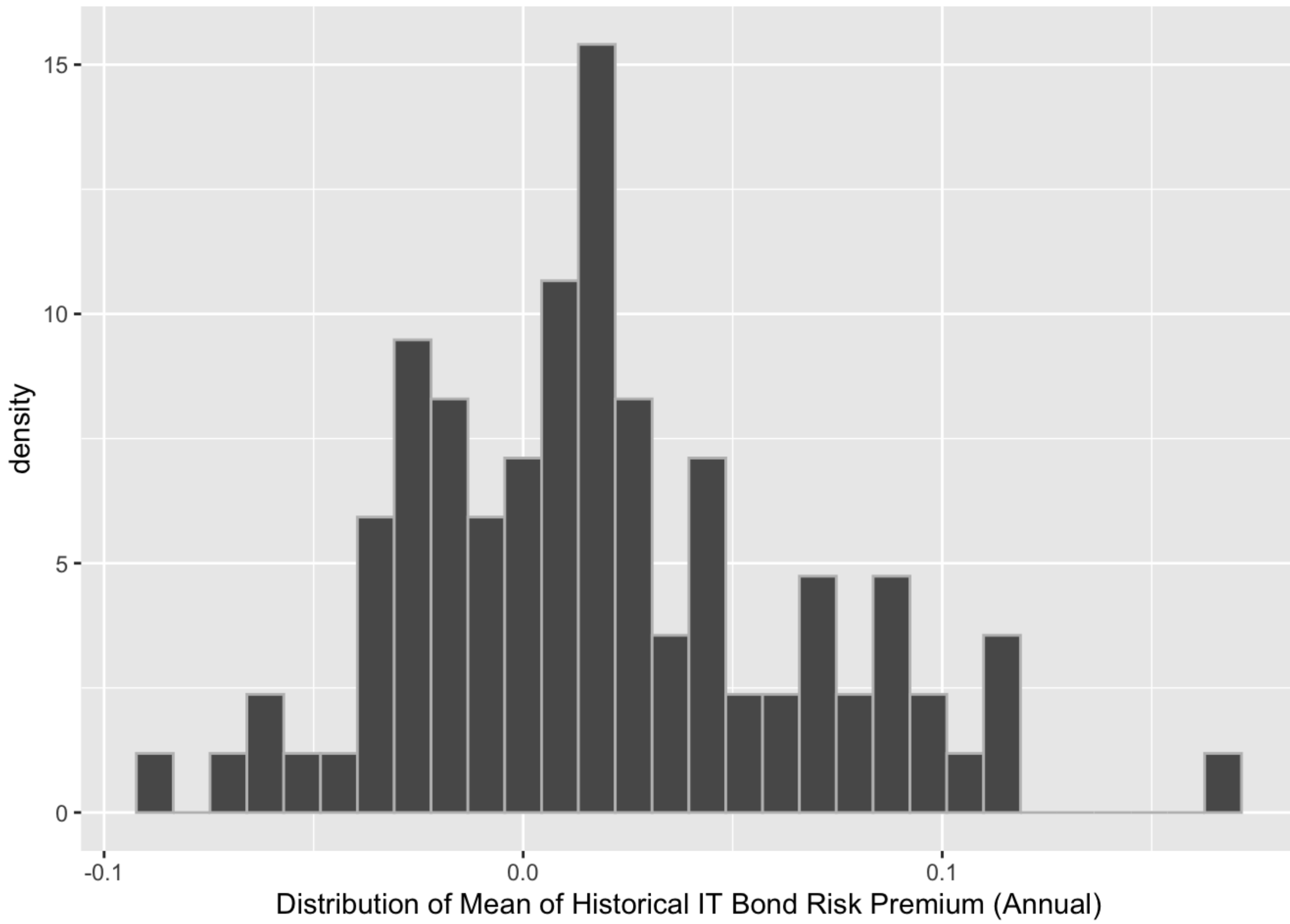
Summary of the overall distribution

```
summary(sbbi_it_bond$sbbi_it_bond_risk_premium)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.08707 -0.01573  0.01372  0.01823  0.04440  0.16784
```

```
sbbi_it_bond %>%
  ggplot(aes(x=sbbi_it_bond_risk_premium))+
  geom_histogram(aes(y=..density..),color = "grey")+
  labs(x = "Distribution of Mean of Historical IT Bond Risk Premium (Annual)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Bucket based on inflation

```
sbbi_it_bond <- sbbi_it_bond[order(sbbi_it_bond$cpi),]
sum(sbbi_it_bond$cpi <=0)
sum((sbbi_it_bond$cpi >0) & (sbbi_it_bond$cpi<=0.025))
sum((sbbi_it_bond$cpi >0.025) & (sbbi_it_bond$cpi<=0.05))
sum(sbbi_it_bond$cpi >=0.05)
sbbi_it_bond <- sbbi_it_bond %>%
  mutate(bucket = c(rep("Deflation",10),
                    rep("Low",35),
                    rep("Medium",33),
                    rep("High",18)))
#sbbi_it_bond <- sbbi_it_bond[order(sbbi_it_bond$obs_date),]
```

Mean of Equity Return within Inflation Bucket

```
sbbi_brp_mean <- sbbi_it_bond %>%
    group_by(bucket) %>%
    summarise(sbbi_brp_mean = mean(sbbi_it_bond_risk_premium))
sbbi_brp_mean <- sbbi_brp_mean[c(1,3,4,2),]
sbbi_brp_mean
```

```
## # A tibble: 4 × 2
##   bucket    sbbi_brp_mean
##   <chr>             <dbl>
## 1 Deflation        0.0232
## 2 Low              0.0259
## 3 Medium           0.0257
## 4 High            -0.0130
```

SBBI - Long-Term Government Bond (20Yr)

```
sbbi_lt_bond <- read_excel("/Users/zexing/Desktop/296B Final/Historical Data/SBBI Data f
or Capital Markets (2022_12_31).xlsx", sheet = "lt_bond") %>%
  rename("obs_date" = 1, "sbbi_lt_bond" = 2,
         "ust_1m" = 3, "sbbi_lt_bond_risk_premium" = 4, "cpi" = 5) %>%
  mutate(obs_date = as.Date(obs_date,origin = "1899-12-30"))
```

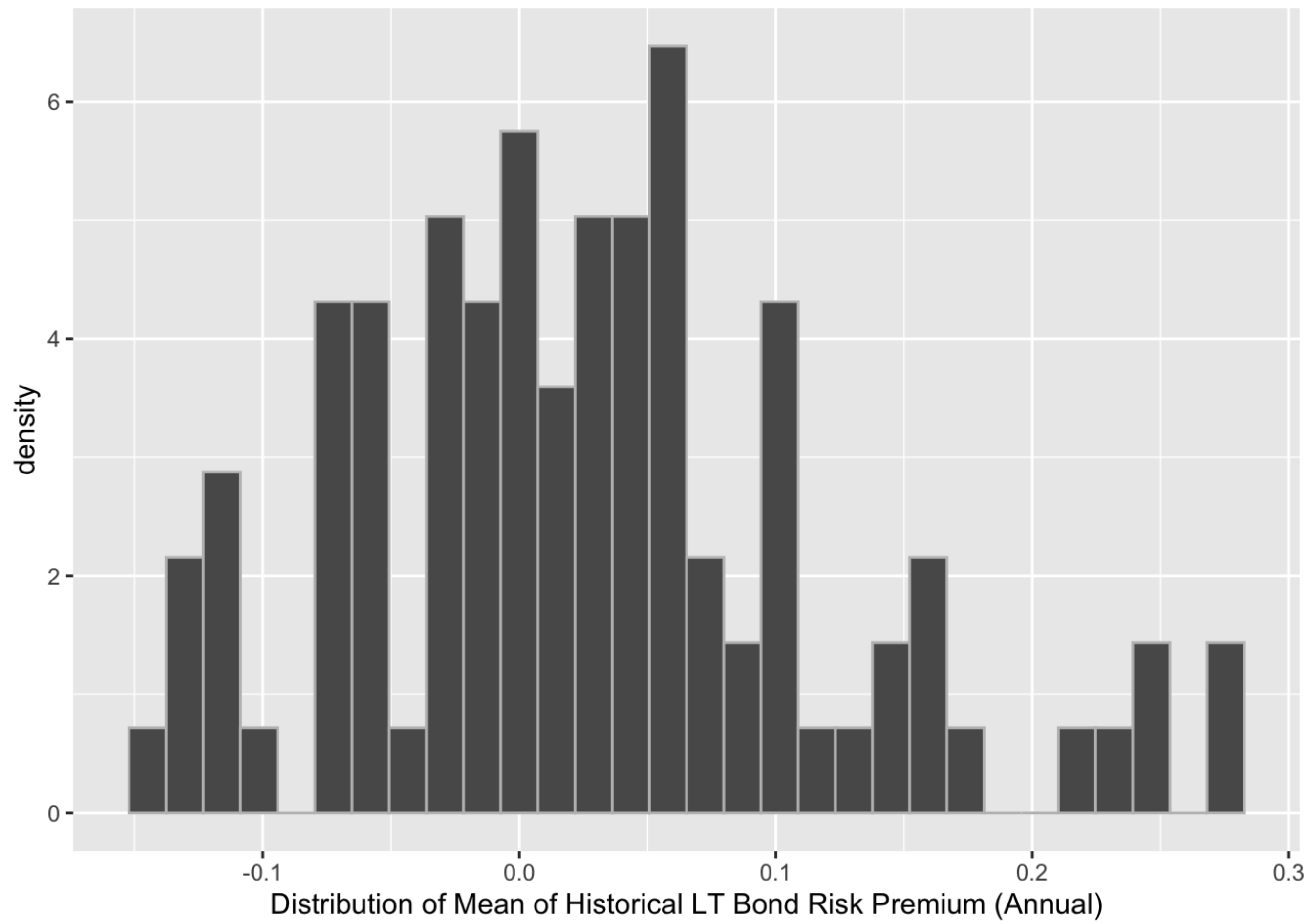
Summary of the overall distribution

```
summary(sbbi_lt_bond$sbbi_lt_bond_risk_premium)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.14986 -0.03346  0.01988  0.02621  0.07246  0.27048
```

```
sbbi_lt_bond %>%
  ggplot(aes(x=sbbi_lt_bond_risk_premium))+
  geom_histogram(aes(y=..density..),color = "grey")+
  labs(x = "Distribution of Mean of Historical LT Bond Risk Premium (Annual)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Bucket based on inflation

```
sbbi_lt_bond <- sbbi_lt_bond[order(sbbi_lt_bond$cpi),]
sum(sbbi_lt_bond$cpi <=0)
sum((sbbi_lt_bond$cpi >0) & (sbbi_lt_bond$cpi<=0.025))
sum((sbbi_lt_bond$cpi >0.025) & (sbbi_lt_bond$cpi<=0.05))
sum(sbbi_lt_bond$cpi >=0.05)
sbbi_lt_bond <- sbbi_lt_bond %>%
  mutate(bucket = c(rep("Deflation",10),
                   rep("Low",35),
                   rep("Medium",33),
                   rep("High",18)))
#sbbi_lt_bond <- sbbi_lt_bond[order(sbbi_lt_bond$obs_date),]
```

Mean of Equity Return within Inflation Bucket

```
sbbi_brp_mean <- sbbi_lt_bond %>%
    group_by(bucket) %>%
    summarise(sbbi_brp_mean = mean(sbbi_lt_bond_risk_premium))
sbbi_brp_mean <- sbbi_brp_mean[c(1,3,4,2),]
sbbi_brp_mean
```

```
## # A tibble: 4 × 2
##   bucket    sbbi_brp_mean
##   <chr>             <dbl>
## 1 Deflation        0.0413
## 2 Low              0.0458
## 3 Medium           0.0380
## 4 High            -0.0418
```

NAIC - 100 Years of Data(Given)

Intermediate Bond Return (5Yr)

```
naic_int_govt <- read.csv("/Users/zexing/Desktop/296B Final/NAIC/naic_int_govt_tr.csv")
naic_int_govt <- naic_int_govt[,-1]

naic_ust_1m <-read.csv("/Users/zexing/Desktop/296B Final/NAIC/naic_ust_1m.csv")
naic_ust_1m <- naic_ust_1m[,-1]
```

Convert to Annual Data

```
naic_int_govt_annual <- data.frame(matrix(1000, ncol=1, nrow = 1000))
for (i in 1:1200){
  naic_int_govt_annual[,i+1] <- naic_int_govt_annual[,i]*(1+naic_int_govt[,i])
}
holder <- data.frame(matrix(0, ncol=100, nrow = 1000))
for (i in 1:100){
  holder[,i] <- naic_int_govt_annual[,1+12*i]/naic_int_govt_annual[,(1+12*(i-1))]-1
}
naic_int_govt_annual <- holder
```

```
summary(sbbi_it_bond$sbbi_it_bond)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.05144  0.01635  0.03698  0.05173  0.07833  0.29097
```

```
summary(unlist(naic_int_govt_annual))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.244652 -0.008385  0.031189  0.044393  0.091631  0.631780
```

```
naic_ust_1m_annual <- data.frame(matrix(1000, ncol=1, nrow = 1000))
for (i in 1:1200){
  naic_ust_1m_annual[,i+1] <- naic_ust_1m_annual[,i]*(1+naic_ust_1m[,i]/12)
}
holder <- data.frame(matrix(0, ncol=100, nrow = 1000))
for (i in 1:100){
  holder[,i] <- naic_ust_1m_annual[,1+12*i]/naic_ust_1m_annual[,(1+12*(i-1))]-1
}
naic_ust_1m_annual <- holder
```

```
summary(unlist(sbbi_it_bond$ust_1m))
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0001622  0.0049023  0.0281529  0.0330392  0.0514161  0.1470886
```

```
summary(unlist(naic_ust_1m_annual))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.011422  0.001234  0.015443  0.030501  0.047900  0.430048
```

Then calculate the Bond Risk Premium.

```
naic_it_brp_annual <- data.frame(matrix(0, ncol=100, nrow = 1000))
for (col in 1:100){
  for (row in 1:1000){
    naic_it_brp_annual[row,col] = (1+naic_int_govt_annual[row,col])/(1+naic_ust_1m_annua
l[row,col]) - 1
  }
}
summary(unlist(naic_it_brp_annual))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.31301 -0.03278  0.01128  0.01394  0.05854  0.49001
```

```
x <- unlist(naic_it_brp_annual)
x<- sbbi_it_bond$sbbi_it_bond_risk_premium
quantile(x,0.01)
```

```
##          1%
## -0.06960217
```

```
quantile(x,0.05)
```

```
##          5%
## -0.04858267
```

```
quantile(x,0.1)
```

```
##         10%
## -0.03301962
```

```
quantile(x,0.9)
```

```
##       90%
## 0.0861061
```

```
quantile(x,0.95)
```

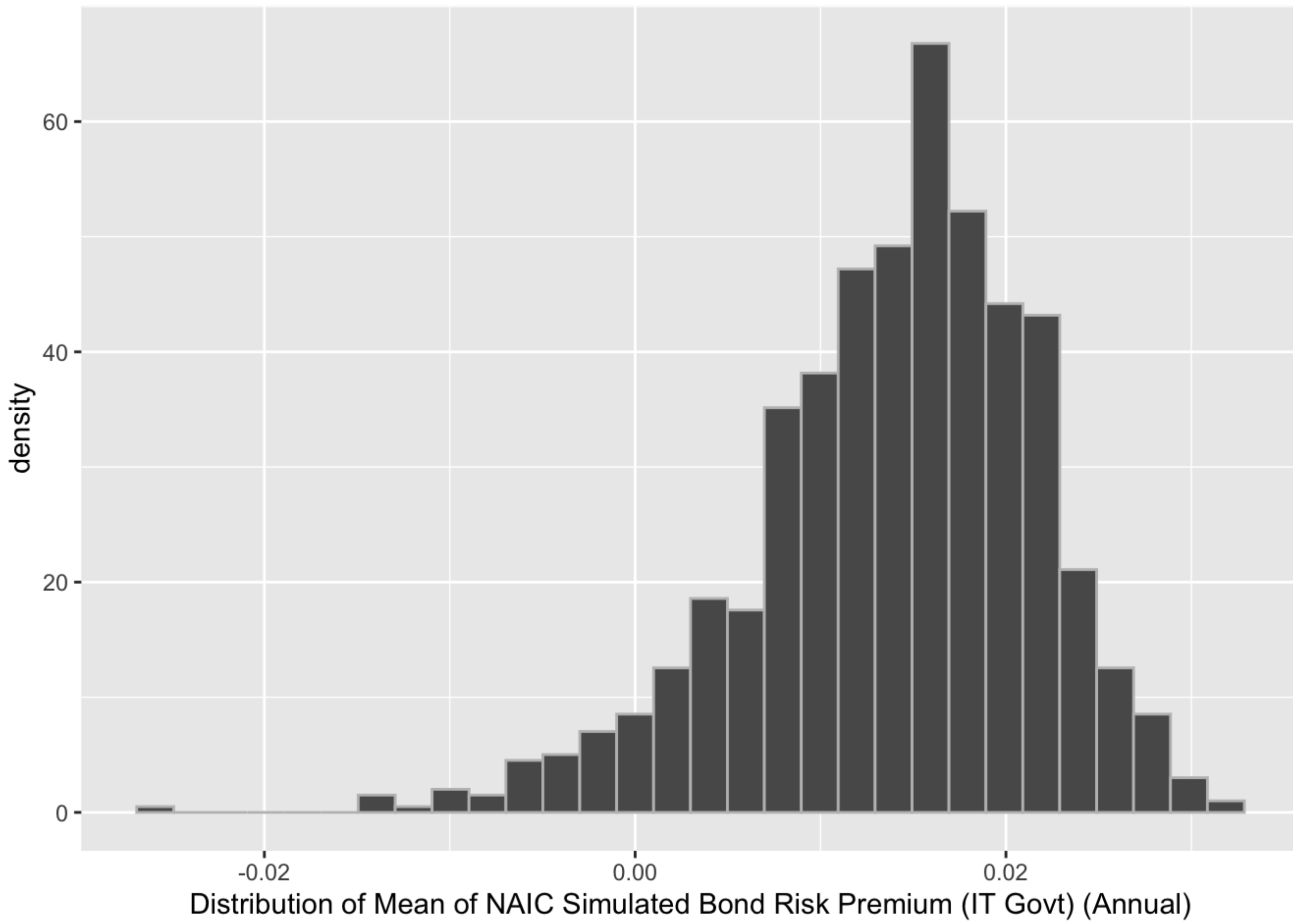
```
##        95%
## 0.09928738
```

```
quantile(x,0.99)
```

```
##       99%
## 0.1195805
```

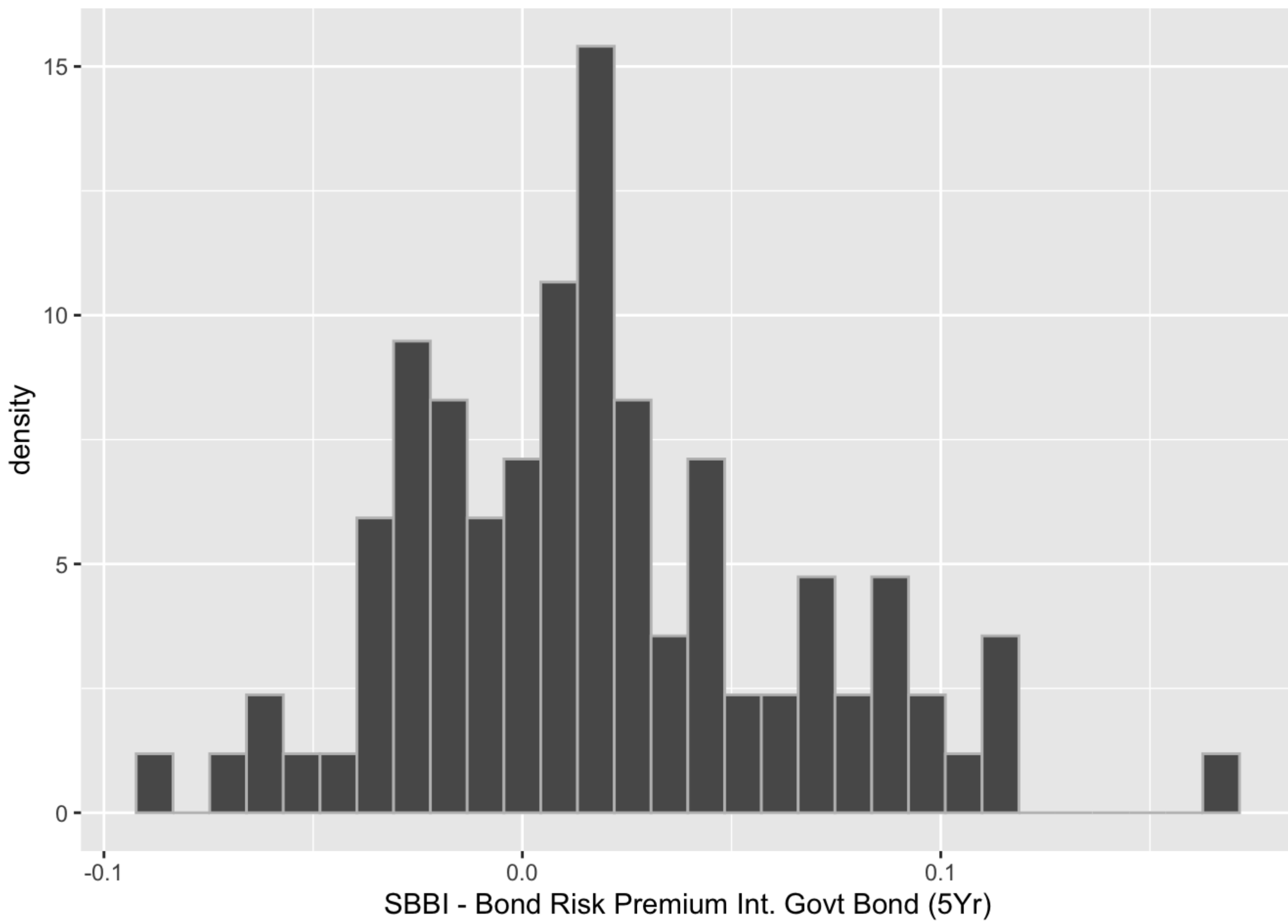
```
mean_container <- numeric(1000)
for (i in 1:1000){
  mean_container[i] = mean(t(as.tibble(naic_it_brp_annual[i,])))
}
mean_container <- as.tibble(mean_container)
mean_container %>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
    labs(x = "Distribution of Mean of NAIC Simulated Bond Risk Premium (IT Govt) (Annua
l)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



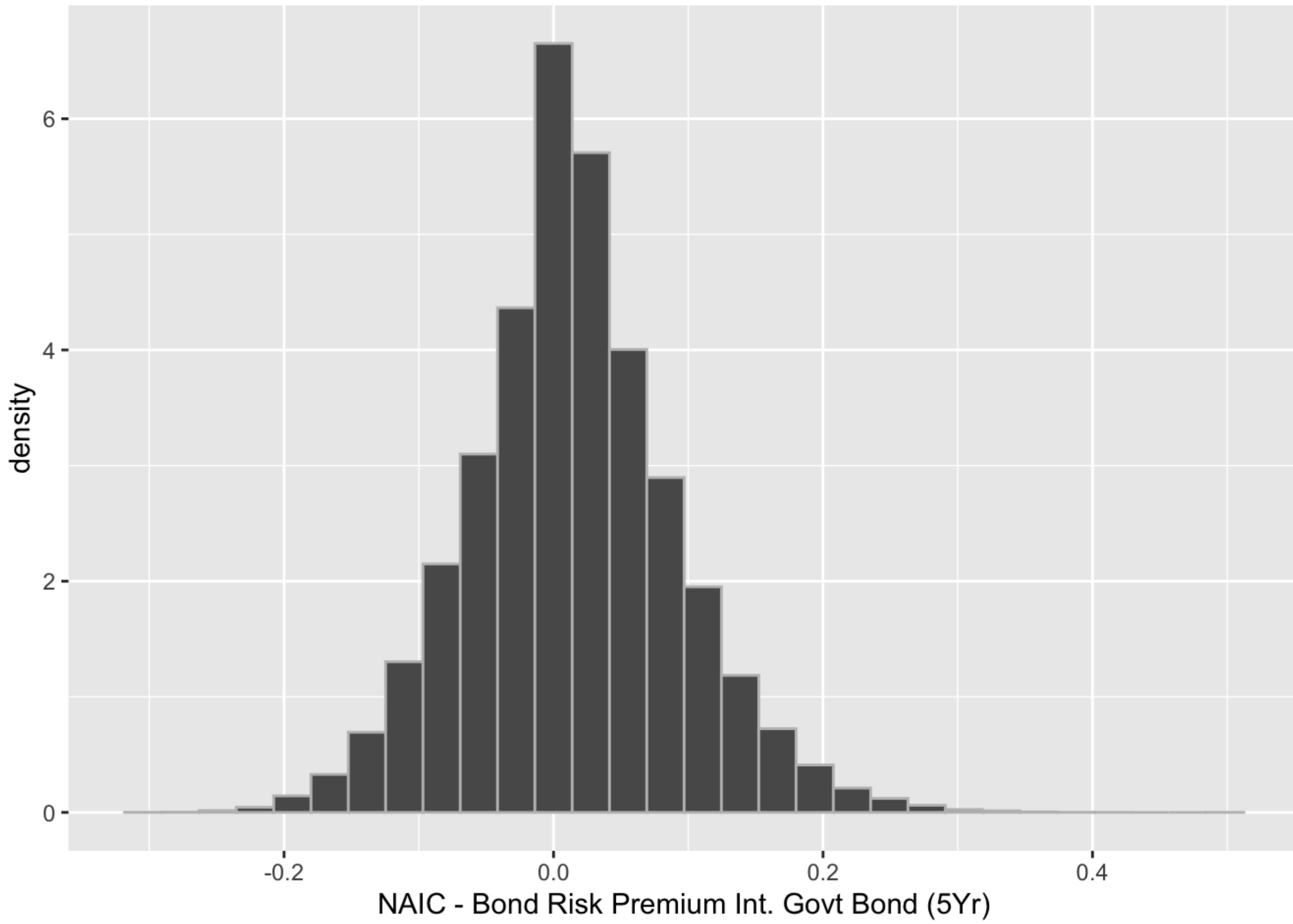

```
sbbi_it_bond %>%
  ggplot(aes(x=sbbi_it_bond_risk_premium)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "SBBI - Bond Risk Premium Int. Govt Bond (5Yr)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
naic_it_brp_annual_v <- as.tibble(unlist(naic_it_brp_annual))
naic_it_brp_annual_v%>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "NAIC - Bond Risk Premium Int. Govt Bond (5Yr)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



LongTerm Bond Return (20Yr)

```
naic_lt_govt <- read.csv("/Users/zexing/Desktop/296B Final/NAIC/naic_lt_govt_tr.csv")
naic_lt_govt <- naic_lt_govt[,-1]

naic_ust_1m <-read.csv("/Users/zexing/Desktop/296B Final/NAIC/naic_ust_1m.csv")
naic_ust_1m <- naic_ust_1m[,-1]
```

Convert to Annual Data

```
naic_lt_govt_annual <- data.frame(matrix(1000, ncol=1, nrow = 1000))
for (i in 1:1200){
  naic_lt_govt_annual[,i+1] <- naic_lt_govt_annual[,i]*(1+naic_lt_govt[,i])
}
holder <- data.frame(matrix(0, ncol=100, nrow = 1000))
for (i in 1:100){
  holder[,i] <- naic_lt_govt_annual[,1+12*i]/naic_lt_govt_annual[,(1+12*(i-1))]-1
}
naic_lt_govt_annual <- holder
```

```
summary(sbbi_lt_bond$sbbi_lt_bond)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.149033 -0.001414  0.036736  0.059712  0.100554  0.403613
```

```
summary(unlist(naic_lt_govt_annual))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.37628 -0.03692  0.03755  0.04862  0.12340  0.85867
```

```
naic_ust_1m_annual <- data.frame(matrix(1000, ncol=1, nrow = 1000))
for (i in 1:1200){
  naic_ust_1m_annual[,i+1] <- naic_ust_1m_annual[,i]*(1+naic_ust_1m[,i]/12)
}
holder <- data.frame(matrix(0, ncol=100, nrow = 1000))
for (i in 1:100){
  holder[,i] <- naic_ust_1m_annual[,1+12*i]/naic_ust_1m_annual[,(1+12*(i-1))]-1
}
naic_ust_1m_annual <- holder
```

```
summary(sbbi_lt_bond$ust_1m)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0001622  0.0049023  0.0281529  0.0330392  0.0514161  0.1470886
```

```
summary(unlist(naic_ust_1m_annual))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.011422  0.001234  0.015443  0.030501  0.047900  0.430048
```

Then calculate the Bond Risk Premium

```
naic_lt_brp_annual <- data.frame(matrix(0, ncol=100, nrow = 1000))
for (col in 1:100){
  for (row in 1:1000){
    naic_lt_brp_annual[row,col] = (1+naic_lt_govt_annual[row,col])/(1+naic_ust_1m_annual
[row,col]) - 1
  }
}
summary(unlist(naic_lt_brp_annual))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.41343 -0.06257  0.01215  0.01819  0.09136  0.75509
```

```
x<- sbbi_lt_bond$sbbi_lt_bond_risk_premium
quantile(x,0.01)
```

```
##         1%
## -0.1371549
```

```
quantile(x,0.05)
```

```
##          5%
## -0.1149284
```

```
quantile(x,0.1)
```

```
##         10%
## -0.0782546
```

```
quantile(x,0.9)
```

```
##       90%
## 0.1532039
```

```
quantile(x,0.95)
```

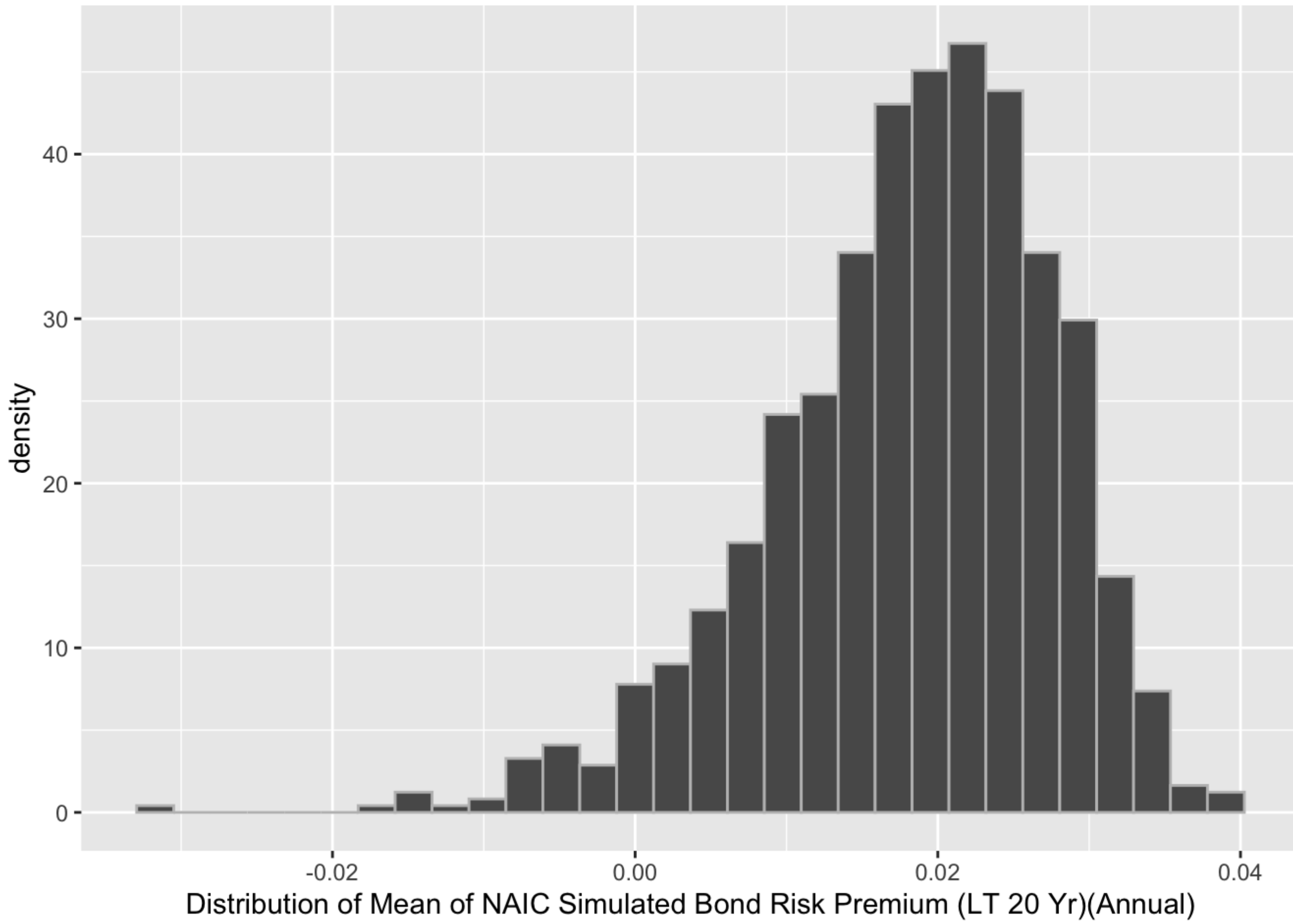
```
##      95%
## 0.221557
```

```
quantile(x,0.99)
```

```
##       99%
## 0.2697799
```

```
mean_container <- numeric(1000)
for (i in 1:1000){
  mean_container[i] = mean(t(as.tibble(naic_lt_brp_annual[i,])))
}
mean_container <- as.tibble(mean_container)
mean_container %>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
    labs(x = "Distribution of Mean of NAIC Simulated Bond Risk Premium (LT 20 Yr)(Annua
l)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



AIRG - 69 Years of Data 1954.1 - 2022.12

AIRG IT Govt Bond TR (Annual)

```
airg_it_govt <- read.csv("/Users/zexing/Desktop/296B Final/AIRG/airg_it_govt_annual.cs
v",header = FALSE)
holder <- data.frame(matrix(0, ncol=70, nrow = 1000))
for (col in 1:70){
  holder[,col] = airg_it_govt[,col+1]/airg_it_govt[,col]-1
}
airg_it_govt <- holder[,-70] #take out the last year bc I simulate 70 years of data, but
realize only need 69
```

UST - 3M (Annual)

```
airg_ust_3m <- read.csv("/Users/zexing/Desktop/296B Final/AIRG/airg_ust_3m_annual.csv",
                          header = FALSE)
airg_ust_3m <-  airg_ust_3m[,-c(1,71)]
```

Then calculate the Bond Risk Premium. The formula is (1+airg_it_govt)/(1+airg_ust_3m)- 1

```r
airg_it_brp_annual <- data.frame(matrix(0, ncol=69, nrow = 1000))
for (col in 1:69){
  for (row in 1:1000){
    airg_it_brp_annual[row,col] = (1+airg_it_govt[row,col])/(1+airg_ust_3m[row,col]) - 1
  }
}
summary(unlist(airg_it_brp_annual))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.259437 -0.016228  0.006910  0.007478  0.030389  0.286574
```

```r
x<- unlist(airg_it_brp_annual)
quantile(x,0.01)
```

```
##          1%
## -0.08153176
```

```r
quantile(x,0.05)
```

```
##          5%
## -0.05096016
```

```r
quantile(x,0.1)
```

```
##         10%
## -0.03727791
```

```r
quantile(x,0.9)
```

```
##        90%
## 0.05333282
```

```r
quantile(x,0.95)
```

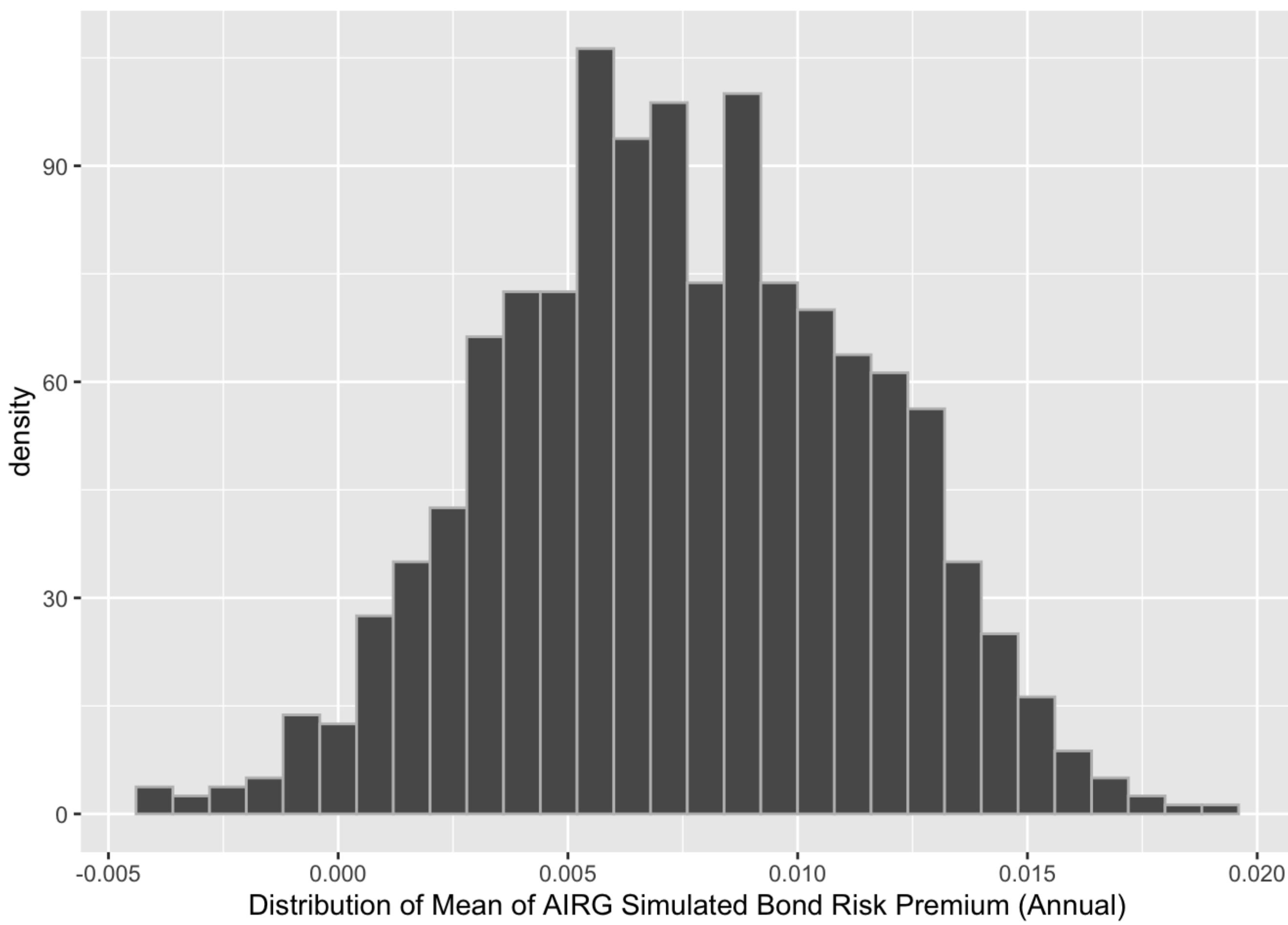
```
##        95%
## 0.06787241
```

```r
quantile(x,0.99)
```

```
##       99%
## 0.1017439
```

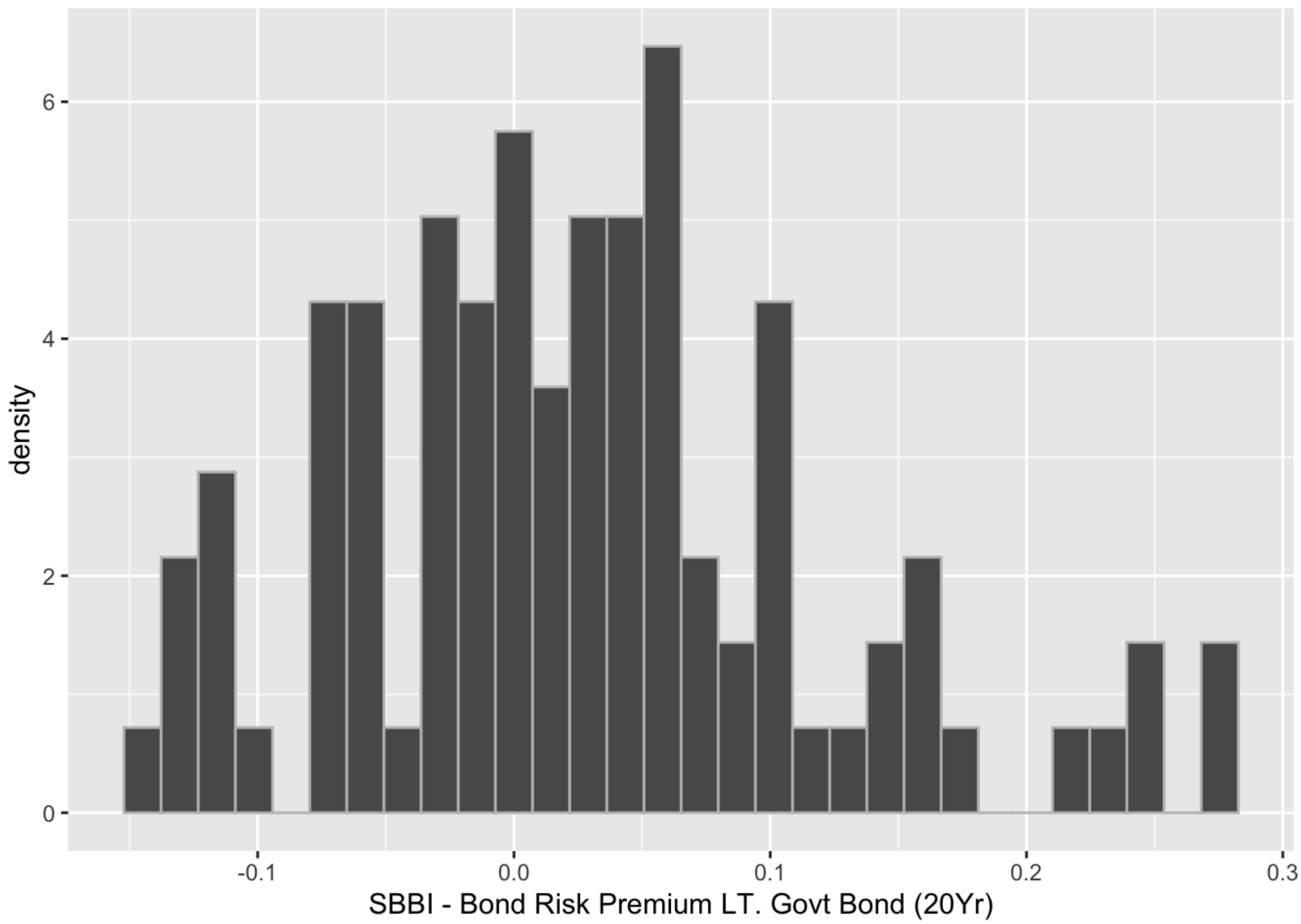
```r
mean_container <- numeric(1000)
for (i in 1:1000){
  mean_container[i] = mean(t(as.tibble(airg_it_brp_annual[i,])))
}
mean_container <- as.tibble(mean_container)
mean_container %>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
    labs(x = "Distribution of Mean of AIRG Simulated Bond Risk Premium (Annual)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



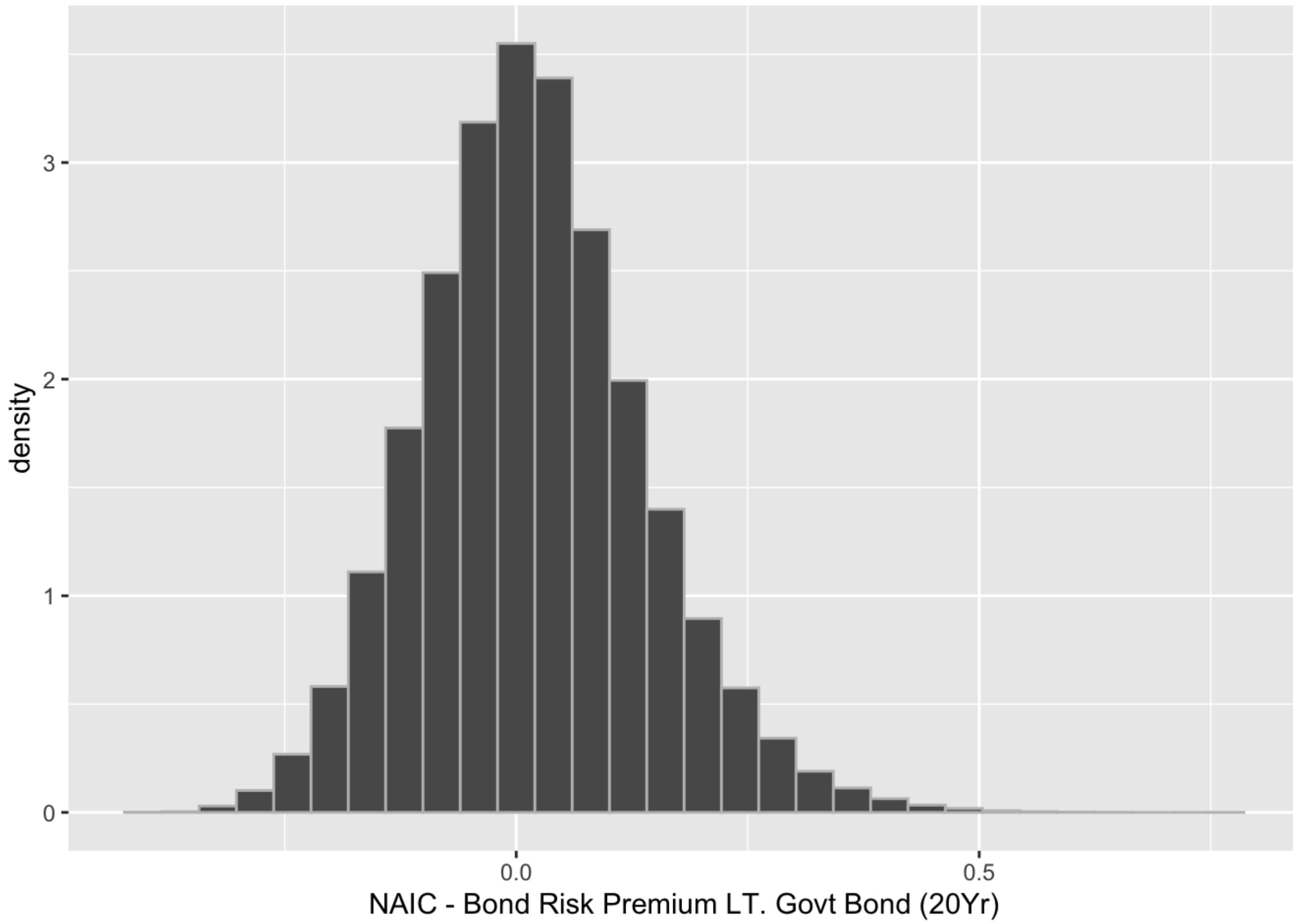

```r
sbbi_lt_bond %>%
  ggplot(aes(x=sbbi_lt_bond_risk_premium)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "SBBI - Bond Risk Premium LT. Govt Bond (20Yr)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



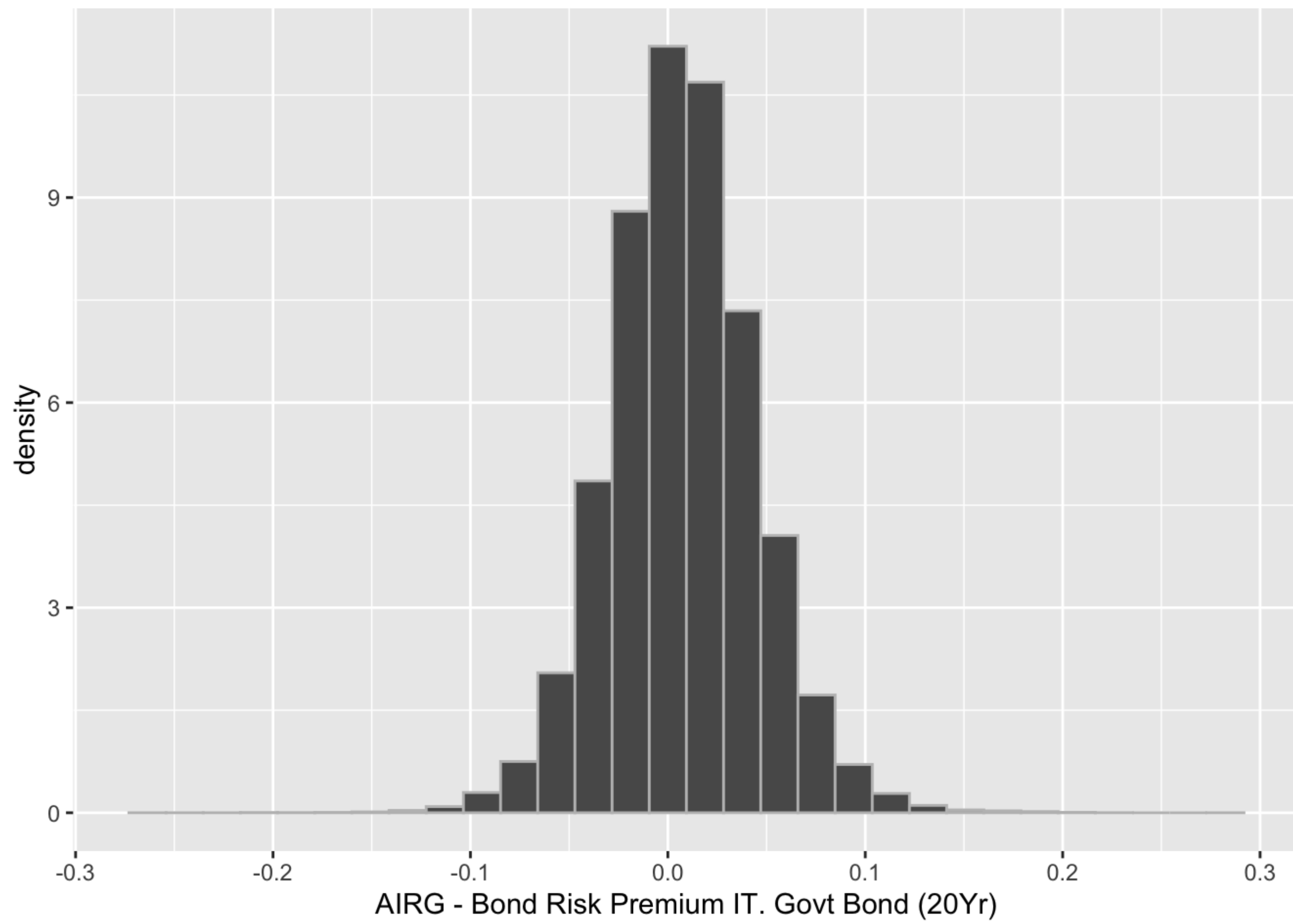

```
naic_lt_brp_annual_v <- as.tibble(unlist(naic_lt_brp_annual))
naic_lt_brp_annual_v%>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "NAIC - Bond Risk Premium LT. Govt Bond (20Yr)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
airg_it_brp_annual_v <- as.tibble(unlist(airg_it_brp_annual))
airg_it_brp_annual_v%>%
  ggplot(aes(x=value)) +
  geom_histogram(aes(y=..density..),color = "grey") +
  labs(x = "AIRG - Bond Risk Premium IT. Govt Bond (20Yr)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



density
9
6
3
0
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
AIRG - Bond Risk Premium IT. Govt Bond (20Yr)