Analysis

Joe Marlo

12/20/2019

Analysis of Social Security benefits

This document analyses the best strategy to claim Social Security. It calculates Social Security benefits for a given wage, forecasts those benefits into the future, and then discounts those benefits back into real dollars at age 62 (the earliest age to claim). If one chooses to invest their benefits, the benefits will be grown with a rate of return. Since individual Social Security benefits discontinue at time of death the benefits stop at the expected longevity (life expectancy) of the person. Given these variables of inflation, investment return, and expected longevity, there exists an optimal age to claim one's benefits.

Source the `SS_calculator.R` which contains the core calculator and `ggplot_themes.R` for the plot formats. This also automatically sources `Helper_functions.R` which contains many of the investment discounting functions.

```
library(tidyverse)
source("SS_calculator.R")
source("Plots/ggplot_themes.R")
```

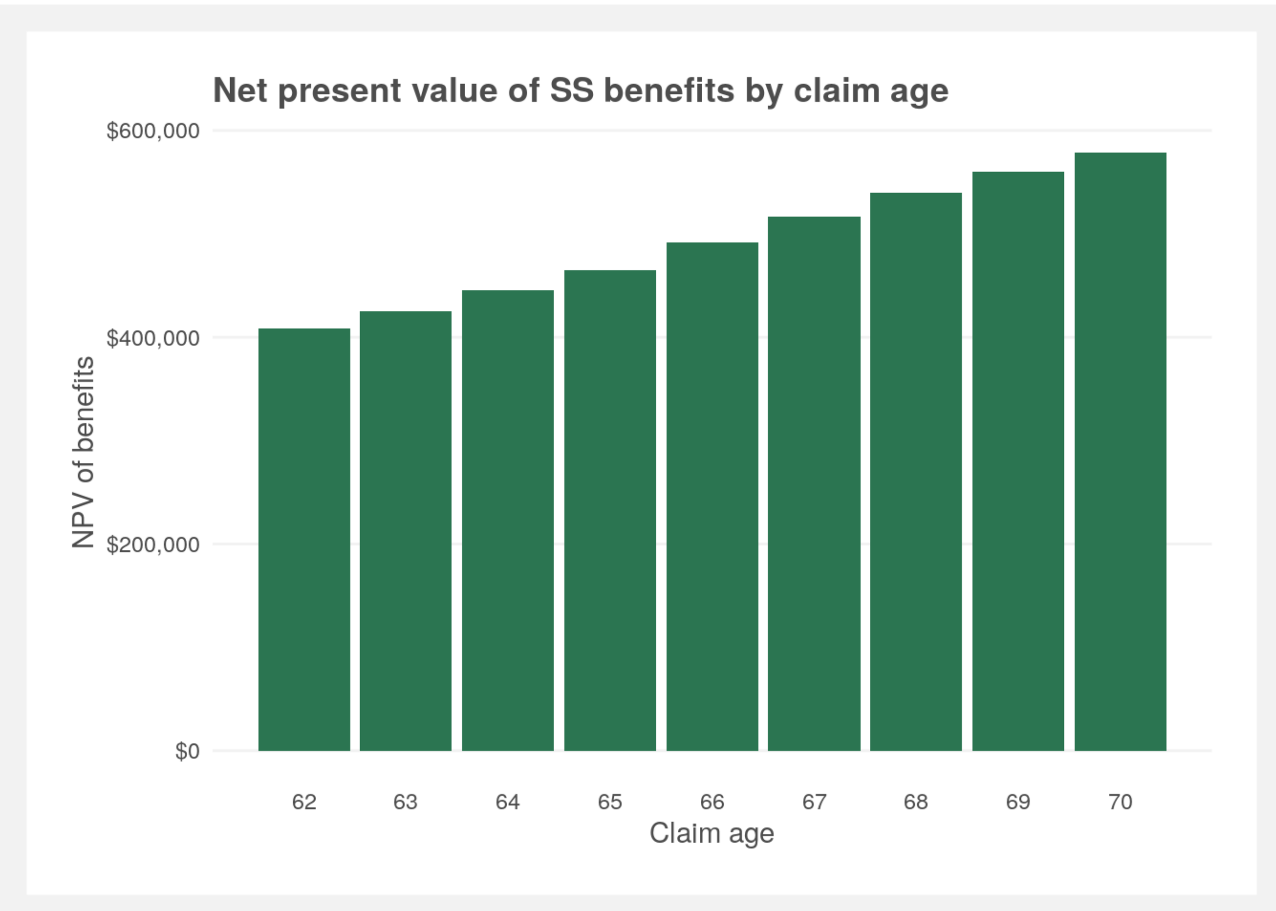
Determining the value of a series of benefits

We can determine to the value of a series benefits by simple net present value discounted at the rate of inflation:

```
# calculating the NPV of a series of annual benefits
benefits <- calculate.benefits(birth.year = 1957, claim.age = 62) %>% filter(Ag
e >= 62)
NPV(benefits$Benefits, inflation.rate)
```

```
## [1] 408793.5
```

```
# plot of NPVs by claim age
sapply(62:70, function(claim.age){
  calculate.benefits(birth.year = 1957, claim.age = claim.age) %>%
    filter(Age >= 62) %>%
    pull(Benefits) %>%
    NPV(., inflation.rate)
}) %>%
  enframe() %>%
  ggplot(aes(x = name, y = value)) +
  geom_col(fill = '#2b7551') +
  scale_x_continuous(breaks = 1:9,
                     labels = 62:70) +
  scale_y_continuous(labels = scales::dollar_format()) +
  labs(title = "Net present value of SS benefits by claim age",
       x = "Claim age",
       y = "NPV of benefits") +
  light.theme
```



However, we want to know is if one invests their benefits and it yields a certain rate of return then how much is that series of benefits worth? We can treat this as an investment account with annual deposits and accumulating growth at a given rate. Then we can take the terminal (final) value and discount it back to age 62. This will give us a single value that is an apples-to-apples comparison across difference claim ages.

```
# calculate the present value of one series of benefits, starting at age 62,
#  ending at age 100, and with an an investment return of 5%
benefits$Benefits %>%
  add.investment.return(., rate = 1.05) %>%
  last(.) %>%
  PV(., rate = inflation.rate, periods = 100-62)
```

```
## [1] 686556.3
```

Great. Now let's see how this present value varies by claim age, investment return, and longevity.

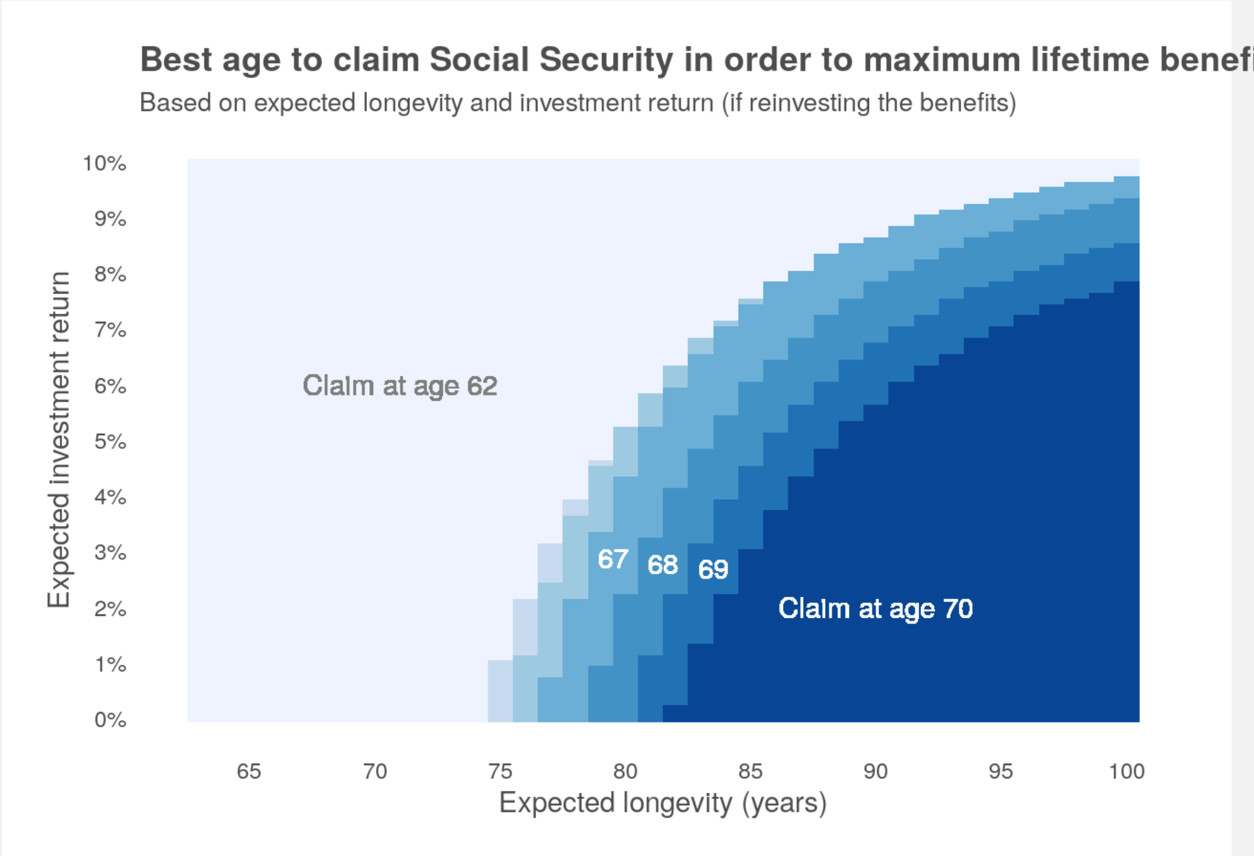
```
# set the investment returns, claim ages, and death ages we are interested in
inv.returns <- seq(1, 1.1, by = 0.001) # this is 0->10%
claim.ages <- 62:70
death.ages <- 63:100

# create all combinations of investment returns, claim age, and death age
PV.grid <- expand.grid(Return = inv.returns,
                       Claim.age = claim.ages,
                       Death.age = death.ages) %>% as_tibble()

# calculate PV for all the combinations
PV.grid$PV <- pmap(list(PV.grid$Return, PV.grid$Claim.age, PV.grid$Death.age),
                   function(inv.return, age, death){

                     # calculate benefits then add investment return then calcu
late present value
                     calculate.benefits(birth.year = 1957, claim.age = age) %>%
                       filter(Age >= 62,
                              Age <= death) %>%
                       pull(Benefits) %>%
                       add.investment.return(., inv.return) %>%
                       last(.) %>%
                       PV(., rate = inflation.rate, periods = death - 62)

                   }) %>% unlist()

head(PV.grid)
```

```
## # A tibble: 6 x 4
##   Return Claim.age Death.age     PV
##    <dbl>     <int>     <int>  <dbl>
## 1   1           62        63 21261.
## 2   1.00        62        63 21271.
## 3   1.00        62        63 21282.
## 4   1.00        62        63 21292.
## 5   1.00        62        63 21303.
## 6   1.00        62        63 21313.
```

We now have present values for every single combination of investment return, claim age, and longevity. What we really care about is the best claim age for a given investment return and longevity. To find this, we can just take the maximum present value for a given grouping of investment return and longevity.

To clearly communicate these results we need a simple plot anyone can use. It should allow the user to pick a life expectancy and expected investment return, go to the intersection and see which claim is best. A simple table will do but that might miss the bigger picture. A tile plot should do the trick.

```
# plot of best claim age by investment return and death age (all claim ages)
PV.grid %>%
  group_by(Return, Death.age) %>%
  filter(PV == max(PV)) %>%
  ggplot(aes(x = Death.age, y = Return, fill = as.factor(Claim.age))) +
  geom_tile() +
  scale_fill_brewer(name = "Claim age") +
  scale_y_continuous(breaks = seq(1, 1.1, by = 0.01),
                     labels = scales::percent(0:10/100, 1)) +
  scale_x_continuous(breaks = seq(65, 100, 5)) +
  geom_text(x = 71, y = 1.06, label = "Claim at age 62", color = "grey50") +
  geom_text(x = 79.5, y = 1.029, label = "67", color = "white") +
  geom_text(x = 81.5, y = 1.028, label = "68", color = "white") +
  geom_text(x = 83.5, y = 1.027, label = "69", color = "white") +
  geom_text(x = 90, y = 1.02, label = "Claim at age 70", color = "white") +
  labs(title = "Best age to claim Social Security in order to maximum lifetime
benefits",
       subtitle = "Based on expected longevity and investment return (if reinve
sting the benefits)",
       x = "Expected longevity (years)",
       y = "Expected investment return") +
  light.theme +
  theme(panel.grid.major.y = element_line(color = NA))
```



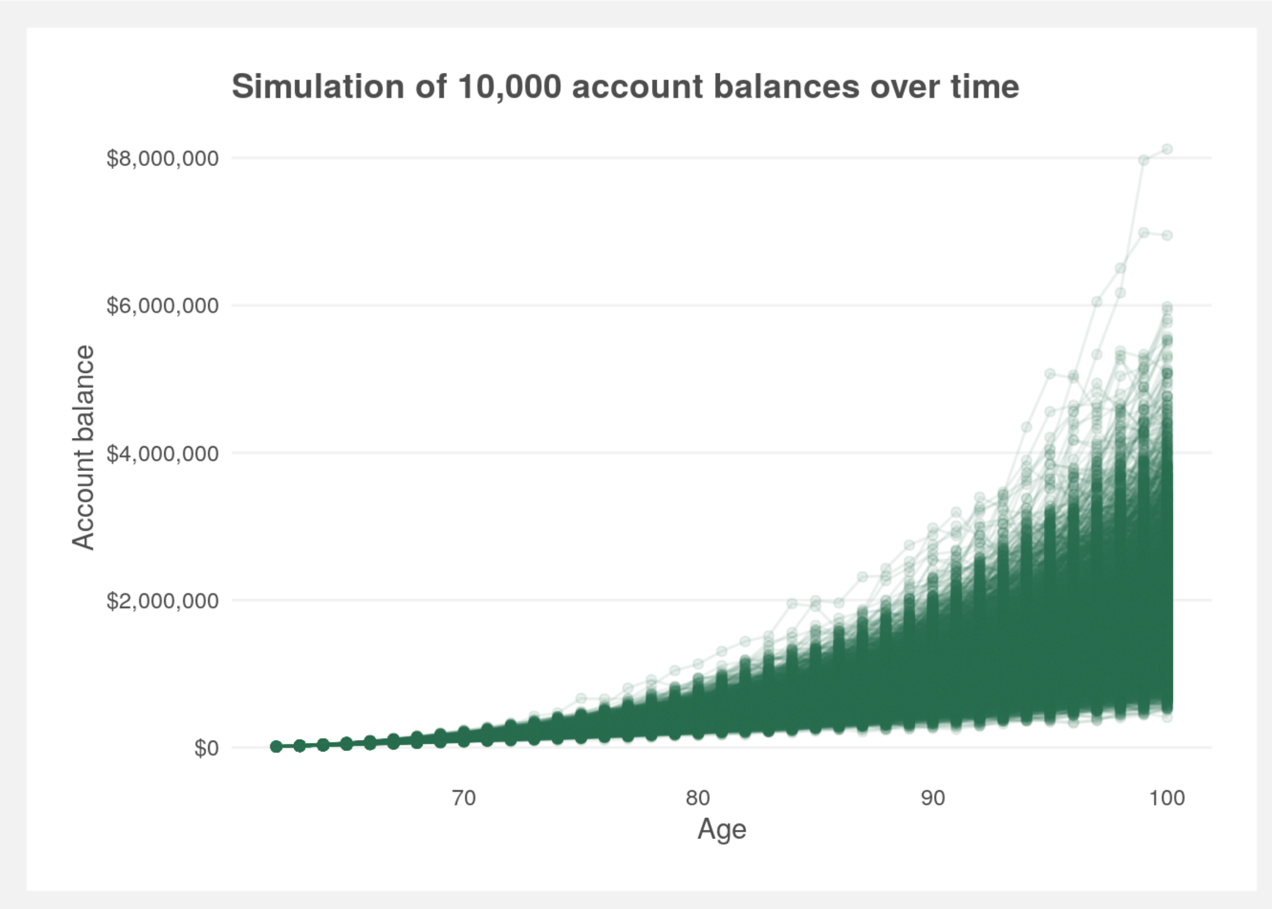

```
ggsave(filename = "Plots/bestClaimAll.png",
       plot = last_plot(),
       device = "png",
       width = 9,
       height = 5)
```

Simulating the investments

Using deterministic investment returns may overestimate final account balances. Simulating (i.e. Monte Carlo) investment returns will most likely reduce the expected rate of return, but understanding that affect usually doesn't require a full Monte Carlo. Where Monte Carlos do make a significant difference is when you have a series of withdrawals or contributions to an account. The variation of investment returns and the timing of those returns tends to have a large effect. Given we are projecting an account balance that has annual contributions, the Monte Carlo method may provide a more accurate prediction than deterministic returns.

Let's simulate a single account first. We can use the already calculated `Benefits` data.frame and apply investment returns using the `sim.investment.return()` function. Doing these many times we can see how the account balances will vary over time – some perform well and others poorly.

```
# set number of simulations, expected return, and expected volatility
n.sims <- 10000
exp.return <- 1.05
exp.vol <- 0.1

# simulate 38 years of account balance 1,000 times
replicate(n.sims,
          sim.investment.return(benefits$Benefits,
                                mean = exp.return,
                                sd = exp.vol)) %>%
  as.data.frame() %>%
  as_tibble() %>%
  mutate(Age = 62:100) %>%
  pivot_longer(cols = starts_with("V")) %>%
  ggplot(aes(x = Age, y = value, group = name)) +
  geom_point(color = "#2b7551", alpha = 0.1) +
  geom_line(color = "#2b7551", alpha = 0.1) +
  scale_y_continuous(labels = scales::dollar_format()) +
  labs(title = paste0("Simulation of ", scales::comma(n.sims), " account balanc
es over time"),
       x = "Age",
       y = "Account balance") +
  light.theme
```

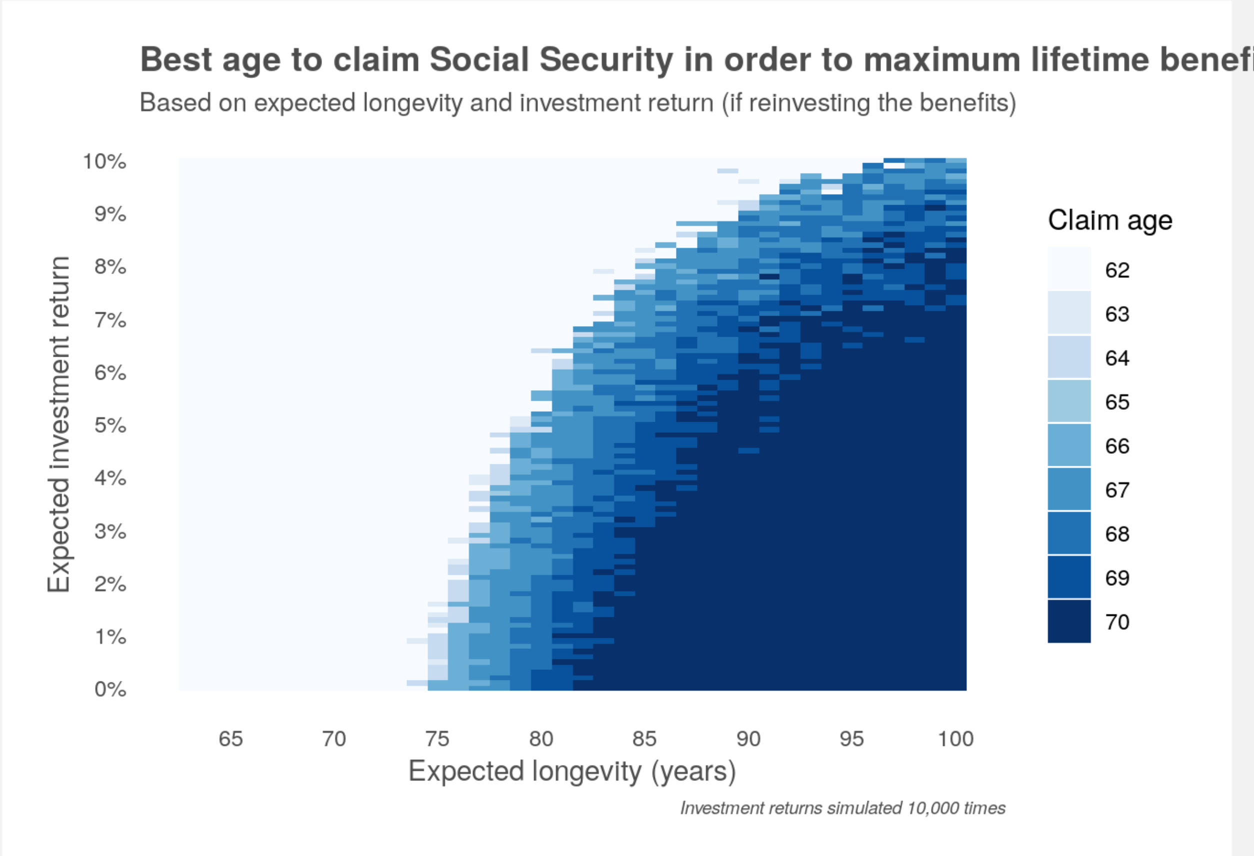


Since we are calculating the present value we only care about the final ending value. And we only need one of those 10,000 values to pick for our present value calculation. We can shoot for the middle of the road by choosing the 50th percentile. A more conservative approach would be choosing the 20th percentile - that represents the value where 80% of the other values are greater than it.

```
# set the percentile (quantile) we are interested in (lower percentile = more c
onservative)
percentile <- 0.50

# calculate benefits then add investment return then calculate present value
benefits$Benefits %>%
  grab.percentile.inv.return(., mean = exp.return, sd = exp.vol,
                             n.sims = n.sims, percentile = percentile) %>%
  PV(., rate = inflation.rate, periods = 100 - 62)
```

```
## [1] 633610.9
```

```
# calculate PV for all the combinations
PV.grid$sim.PV <- pmap(list(PV.grid$Return, PV.grid$Claim.age, PV.grid$Death.ag
e),
                   function(inv.return, age, death){

                     # calculate benefits then add investment return then calcu
late present value
                     calculate.benefits(birth.year = 1957, claim.age = age) %>%
                       filter(Age >= 62,
                              Age <= death) %>%
                       pull(Benefits) %>%
                       grab.percentile.inv.return(., mean = inv.return, sd = ex
p.vol,
                                                  n.sims = n.sims, percentile =
percentile) %>%
                       PV(., rate = inflation.rate, periods = death - 62)
                   }) %>% unlist()

# plot of best claim age by investment return and death age (all claim ages)
PV.grid %>%
  group_by(Return, Death.age) %>%
  filter(sim.PV == max(sim.PV)) %>%
  ggplot(aes(x = Death.age, y = Return, fill = as.factor(Claim.age))) +
  geom_tile() +
  scale_fill_brewer(name = "Claim age") +
  scale_y_continuous(breaks = seq(1, 1.1, by = 0.01),
                     labels = scales::percent(0:10/100, 1)) +
  scale_x_continuous(breaks = seq(65, 100, 5)) +
  labs(title = "Best age to claim Social Security in order to maximum lifetime
benefits",
       subtitle = "Based on expected longevity and investment return (if reinve
sting the benefits)",
       caption = paste("Investment returns simulated", scales::comma(n.sims), "
times"),
       x = "Expected longevity (years)",
       y = "Expected investment return") +
  light.theme +
  theme(panel.grid.major.y = element_line(color = NA),
        plot.caption = element_text(color = "gray30",
                                    face = 'italic',
                                    size = 7),
        legend.position = "right")
```




```
ggsave(filename = "Plots/bestClaimSim.png",
       plot = last_plot(),
       device = "png",
       width = 9,
       height = 5)
```

Conclusion

I find the original deterministic plot much easier to read. The trend is clear and there's obviously less noise. However, I believe the takeaway from the simulated version is to lower you're expections of investment return. Therefore, if you're going to use the deterministic version perhaps it's best to use a lower y value.

An investment return of 5% is probably a safe bet for most people. At that rate and using a life expectancy of 90 (about average for a couple currently age 62), it's best to wait to claim Social Security at age 70.