ITESO, Universidad
Jesuita de Guadalajara

Modelo de ahorro, tasa variable

Pablo Alejandro Rivera Sánchez

Claudio Rodriguez Orozco

4 de diciembre de 2019

Simulación matemática

Carlos Arellano Muro

Objetivos

El objetivo principal de este proyecto es realizar o simular un modelo de ahorro, sin embargo, utilizaremos una tasa de interés variable o aleatoria cada mes en lugar de una tasa fija. De esta manera no sólo podremos ver la cantidad a abonar cada mes, si no el capital final en diferentes escenarios para analizar si es conveniente utilizar el modelo de ahorro. Además, aplicaremos el mismo concepto de interés variable en un problema de cuánto tiempo dejar tu dinero invertido para alcanzar cierto balance al final del periodo de tiempo establecido. El objetivo del proyecto se realizará con la ayuda de los temas vistos en el último parcial de la materia, utilizaremos tanto actividades en clase como tareas para apoyarnos y lograr nuestro objetivo. Además, este proyecto también buscamos reforzar y aplicar los temas vistos en clase en una situación planteada por nosotros.

Problema 1

El primer problema era proyectar un ahorro mensual para que al terminar cierto periodo de tiempo, tener $80000 MXN con un capital inicial de $1000 MXN.

Dicho problema se resuelve de una manera bastante simple, la cual ya vimos brevemente en clase, sin embargo quisimos crear el mismo modelo de ahorro pero en esta ocasión con una tasa de interés variable, esto con el objetivo de que no fuera tan simple y para poder observar diferentes escenarios. A continuación se muestra la resolución con un interés fijo:

In [2]:
```
ci=1000       # Capital inicial
i=0.061       # Tasa de interés
meta=80000    # Meta
an=3          # Años que me faltan para terminar mi carrera
cm=12         # Capital mensual (abonos)
hoy=date(2019,12,1)
```

In [3]:
```
ab=-np.pmt(i/cm,an*cm,meta-ci)
ab
```

Out[3]:
```
2406.9141872106634
```

In [4]:
```
ran=pd.date_range(hoy,freq='MS',periods=an*cm)
ran.name='Fecha de abonos'
```

In [5]:
```
tabla=pd.DataFrame(index=ran,columns=['Abonos','Capital/Balance'],dtype=float)
```

In [6]:
```
tabla['Abonos']=ab
ind=np.arange(1,len(ran)+1)
tabla['Capital/Balance']=ci*(1+i)**ind+ab*(((1+i)**ind-1)/i)
```

In [7]:
```
tabla
```

Out[7]:

Abonos Capital/Balance

Fecha de abonos	Abonos	Capital/Balance
2019-12-01	2406.914187	3467.914187
2020-01-01	2406.914187	6086.371140
2020-02-01	2406.914187	8864.553967
2020-03-01	2406.914187	11812.205946
2020-04-01	2406.914187	14939.664696
2020-05-01	2406.914187	18257.898429
2020-06-01	2406.914187	21778.544421
2020-07-01	2406.914187	25513.949818
2020-08-01	2406.914187	29477.214944
2020-09-01	2406.914187	33682.239242
2020-10-01	2406.914187	38143.770023
2020-11-01	2406.914187	42877.454182
2020-12-01	2406.914187	47899.893074
2021-01-01	2406.914187	53228.700739
2021-02-01	2406.914187	58882.565671
2021-03-01	2406.914187	64881.316365
2021-04-01	2406.914187	71245.990850
2021-05-01	2406.914187	77998.910479
2021-06-01	2406.914187	85163.758206
2021-07-01	2406.914187	92765.661643
2021-08-01	2406.914187	100831.281191
2021-09-01	2406.914187	109388.903531
2021-10-01	2406.914187	118468.540833
2021-11-01	2406.914187	128102.036011
2021-12-01	2406.914187	138323.174395
2022-01-01	2406.914187	149167.802220
2022-02-01	2406.914187	160673.952343
2022-03-01	2406.914187	172881.977623
2022-04-01	2406.914187	185834.692445
2022-05-01	2406.914187	199577.522872
2022-06-01	2406.914187	214158.665954
2022-07-01	2406.914187	229629.258765
2022-08-01	2406.914187	246043.557736
2022-09-01	2406.914187	263459.128946
2022-10-01	2406.914187	281937.049999
2022-11-01	2406.914187	301542.124236

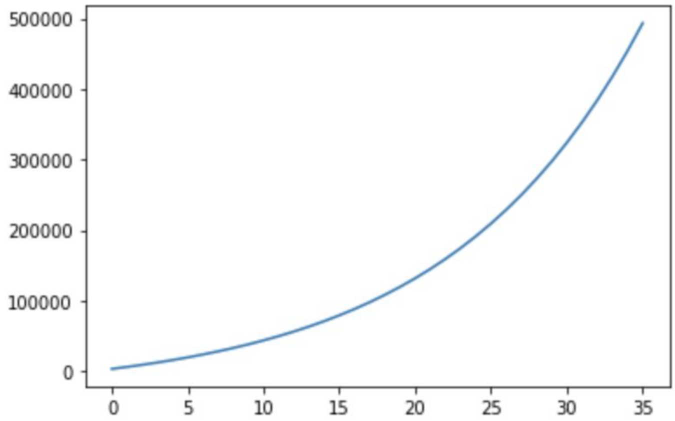
En el ejemplo anterior se utilizó una tasa de interés del 6.1% anual, por un periodo de 3 años o 36 meses. Podemos observar que al final de dicho periodo tendremos \$301,542 si abonamos cada mes la cantidad de \$2,407.

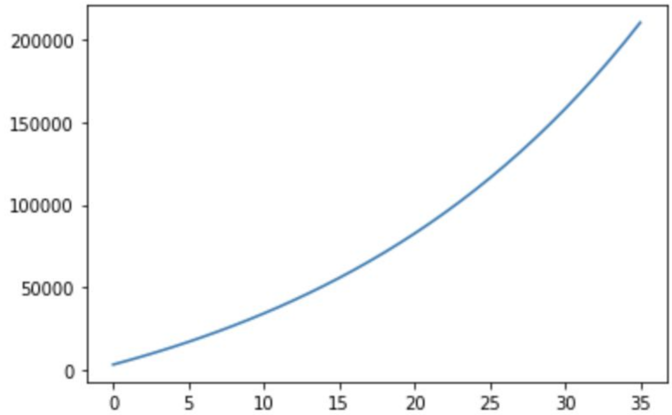
Ahora resolvemos el mismo problema pero con una tasa de interés variable, dicha tasa la obtenemos con la función np.random.normal, a la cual introducimos la tasa de interés anual junto con una volatilidad determinada por nosotros. Utilizamos los mismo datos, más una volatilidad del 2%, para resolver el mismo problema, obtuvimos los siguientes resultados.

```
def inv(ci,i,v,meta,an,cm):
    #ran=pd.date_range(hoy,freq='MS',periods=an*cm)
    #ran.name='Fechas de abonos'
    tabla=pd.DataFrame(index=list(range(an*cm)),columns=['Abonos','Capital/Balance','Tasa'])
    ind=np.arange(1,36+1)
    for w in range(an*cm):
        ir=np.random.normal(i,v)
        ab=-np.pmt(ir/cm,an*cm,meta-ci)
        cp=ci*(1+ir)**ind+ab*(((1+ir)**ind-1)/ir)
        tabla.loc[w,'Abonos'],tabla.loc[w,'Tasa']=ab,ir
        tabla['Capital/Balance']=cp
    return tabla
```

	Abonos	Capital/Balance	Tasa
0	2312.63	3511.459269	0.03436
1	2468.35	6199.901408	0.0780172
2	2574.35	9077.798424	0.106785
3	2391.34	12158.501227	0.0566453
4	2376.19	15456.301568	0.0523896
5	2496.82	18986.498339	0.0858149
6	2400.32	22765.468549	0.0591592
7	2477.18	26810.743294	0.0804408
8	2380.59	31141.089090	0.0536274
9	2414.96	35776.594933	0.0632429
10	2321.69	40738.765492	0.036949
11	2388.47	46050.620874	0.0558387
12	2327.8	51736.803413	0.0386906
13	2392.02	57823.691995	0.0568357
14	2508.5	64339.524429	0.0890003
15	2446.31	71314.528445	0.0719403
16	2399.52	78781.061925	0.0589349
17	2435.04	86773.763017	0.0688231
18	2329.73	95329.710823	0.0392418
19	2369.57	104488.597413	0.0505277
20	2339.32	114292.911965	0.0419691
21	2467.52	124788.137872	0.0777881
22	2527.37	136022.963751	0.0941233
23	2255.08	148049.509311	0.0177708
24	2403.73	160923.567142	0.0601108
25	2394.67	174704.861548	0.0575773
26	2560.12	189457.325612	0.102965
27	2438.83	205249.397789	0.0698724
28	2455.97	222154.339401	0.0746084
29	2389.88	240250.574507	0.0562358
30	2479.14	259622.053718	0.0809779
31	2410.21	280358.643656	0.0619206
32	2495.8	302556.543857	0.0855367
33	2385.16	326318.733050	0.0549099
34	2251.2	351755.446890	0.0166405
35	2440.99	378984.689351	0.0704701

En este caso obtuvimos un capital final superior a la simulación con tasa fija, $379,000, pero como ya sabemos esto puede variar en cada simulación.

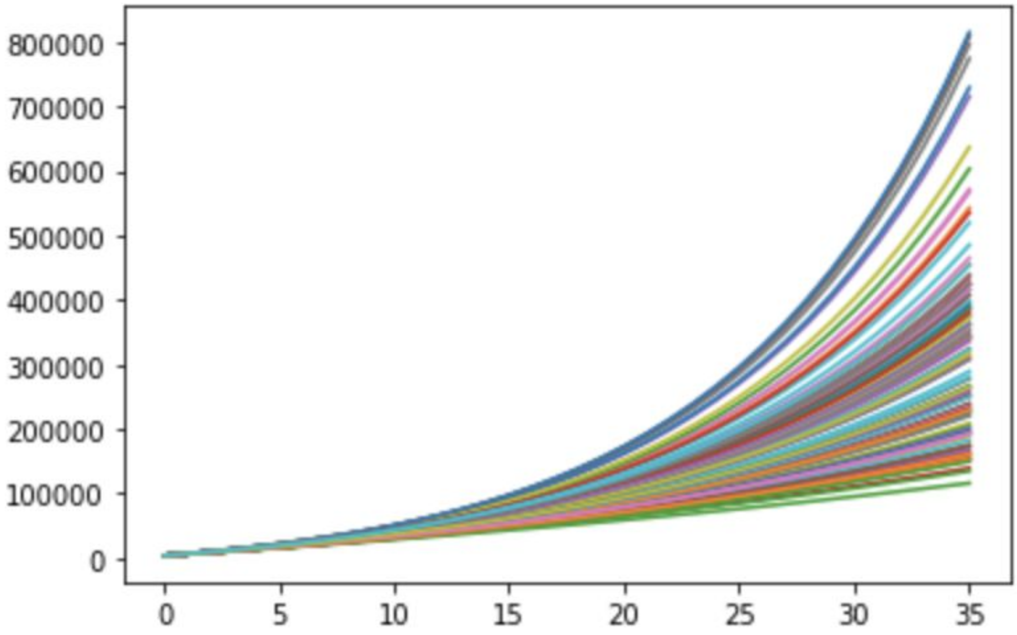




Debido a lo anteriormente mencionado, utilizamos y aplicamos lo visto en el parcial pasado y metimos el proceso a una simulación montecarlo para poder analizar una gran cantidad de resultados o simulaciones.

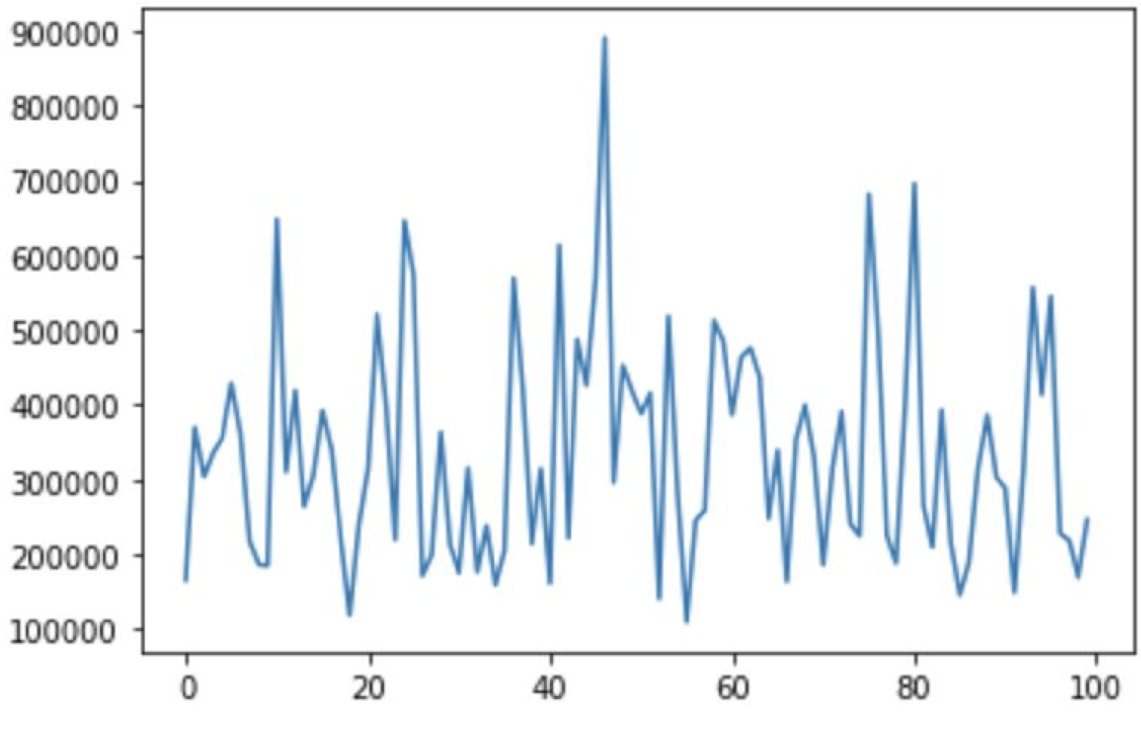
```
def invm(ci,i,v,meta,an,cm):
    #ran=pd.date_range(hoy,freq='MS',periods=an*cm)
    #ran.name='Fechas de abonos'
    tabla=pd.DataFrame(index=list(range(an*cm)),columns=['Abonos','Capital/Balance','Tasa'])
    ind=np.arange(1,36+1)
    for w in range(an*cm):
        ir=np.random.normal(i,v)
        ab=-np.pmt(ir/cm,an*cm,meta-ci)
        cp=ci*(1+ir)**ind+ab*(((1+ir)**ind-1)/ir)
        tabla.loc[w,'Abonos'],tabla.loc[w,'Tasa']=ab,ir
        tabla['Capital/Balance']=cp
    return tabla['Capital/Balance']
```

```
N=100
for i in range(N):
    plt.plot(invm(1000,0.061,0.02,80000,3,12))
```



Nuestro ahorro siempre se verá igual, ya que al variar la tasa de interés también varían nuestros abonos, lo único que cambia es el capital al final de los 3 años el cual también se puede visualizar de la siguiente manera.

```
N=100
x=[]

for w in range(N):
    x.append(invp(1000,0.061,0.02,80000,3,12))
x
```



Esta visualización o tabla nos muestra únicamente los saldos o el capital final en 100 escenarios diferentes, a partir de esto podemos hacer varios análisis como por ejemplo obtener la media de capital final, la cual en este ejemplo es de:

```
336731.70442533866
```

Al hacer la simulación de 100 escenarios o posibilidades varias veces, observamos que la media en general se aproxima a alrededor de $330,000.00.

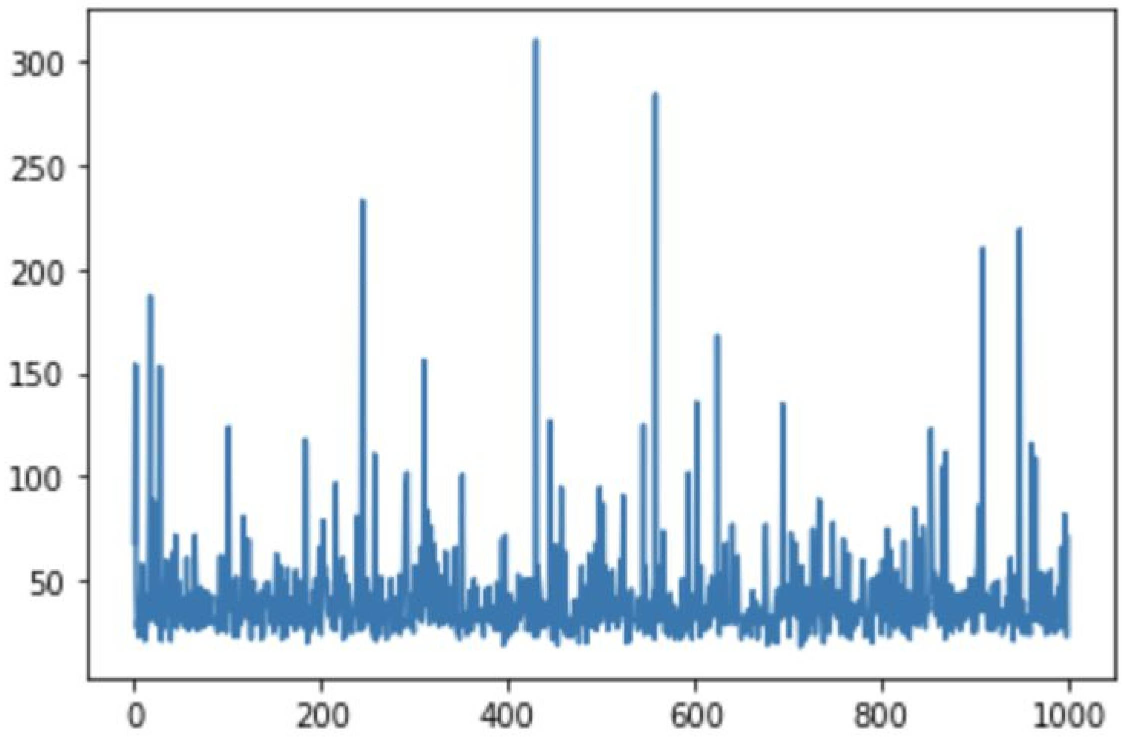
```
333777.72295239085    335143.61887897266

334381.18952703883    322390.35357448185
```

Problema 2

¿Cuánto tiempo (en meses) se debe de dejar invertido el dinero para llegar a $80,000 empezando con $10,000?

Al igual que el problema anterior se utilizará una tasa de interés variable con la misma función de np.random.normal. En este caso la tasa de interés es de 6.1% mientras que la volatilidad es de 2%.

```python
def invk(C0,i,v,meta):
    l=[]
    for w in range(1000):
        ir=np.random.normal(i,v)
        k=np.ceil(np.log(meta/C0)/np.log(1+ir))
        Ck=round(C0*(1+ir)**k,2)
        l.append(k)
    return l
```



En la gráfica podemos observar cuantos meses tardamos en cumplir nuestro objetivo, simulando 1000 escenarios en los que la tasa de interés era variable. En dicho ejemplo obtuvimos una media de 41.16 meses para llegar de $10,000 a $80,000, siendo únicamente 18 meses en el caso más rápido (mínimo) y 310 meses en el caso más lento (máximo).

Conclusiones

Aún no hemos llevado la materia de ecuaciones diferenciales por lo que dicho tema se nos hizo un poco confuso, sin embargo sabemos que tiene varias aplicaciones prácticas que veremos y entenderemos en un futuro. En cuanto al modelo de ahorro, es una herramienta que le puede servir a cualquier persona para analizar lo que un banco le ofrece, nosotros pudimos relacionar este tema con el de simulación montecarlo, visto en el parcial anterior, lo cual fue bastante interesante ya que nuevamente pudimos observar que este tema tiene infinitas aplicaciones si lo sabes plantear. Al principio fue un poco difícil decidir qué íbamos a hacer de proyecto, pero al pensar y ver que podíamos incluir o aplicar un poco de simulación montecarlo en el modelo de ahorro se nos hizo buena idea, además de hacer un problema un poco simple no tan simple gracias a los diferentes escenarios que pudimos simular.

Referencias

- Alpha Bench. (2017). Monte Carlo Simulation with Python. 08/11/19, de Alpha Bench Sitio web: https://alphabench.com/data/monte-carlo-simulation-python.html
- ProyectoModulo3_RiveraP_RodríguezC y códigos de clase/tareas.