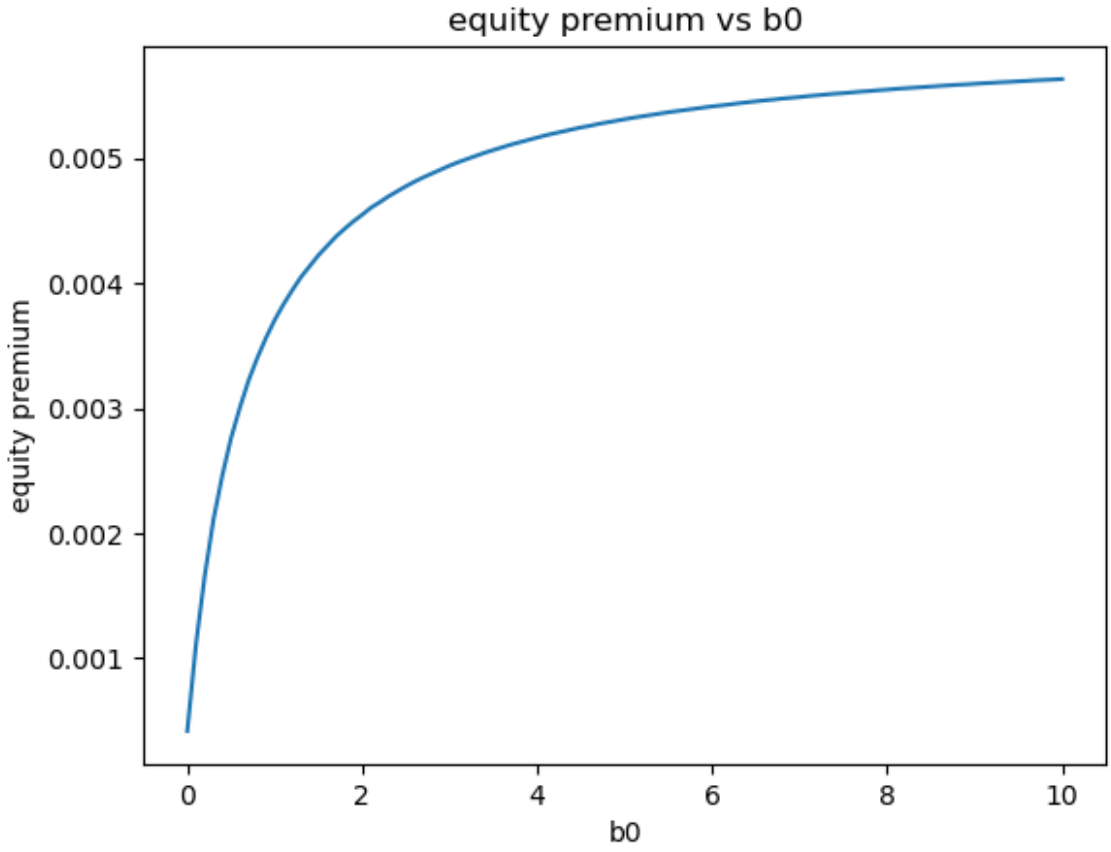
In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:
```
#parameters setting
lng_intercept = 0.02
lng_slope = 0.02
simulate_time = int(1e4)

Rf = 1.0303

penalize_coefficient = 2
##############################
#g = e^(0.02 + 0.02*epsilon)
#lng = 0.02 + 0.02*epsilon
gs = np.exp(lng_intercept+lng_slope*np.random.normal(0,1,simulate_time))

def cal_E(x,gs):
    summation = 0
    for g in gs:
        if x*g >= Rf:
            summation += x*g - Rf
        else:
            summation += penalize_coefficient*(x*g - Rf)
    mean = summation/len(gs)
    return mean

#parameters setting
#e(x) = 0.99*b_0*E(v(x*g)) + 0.99*x - 1
def cal_x(b0, x1, x2):

    x = (x1+x2)/2

    while(abs(0.99*b0*cal_E(x, gs) + 0.99*x - 1))>(1e-5):
        if (0.99*b0*cal_E(x, gs) + 0.99*x - 1)>0:
            x2 = x
        else:
            x1 = x
        x = (x1+x2)/2

    return x
##############################################

#parameters setting
b0_start = 0
b0_end = 10.1
b0_increment = 0.1

x1 = 1
x2 = 1.1

Rm_coef = 0.0202
#####################################################

df_ans = pd.DataFrame([np.arange(b0_start,b0_end,b0_increment)], index=["b0

lst_x = []
for b0 in df_ans["b0"]:
    lst_x.append(cal_x(b0, x1, x2))

df_ans["x"] = lst_x

df_ans["price_dividend_ratio"] = 1/(df_ans["x"]-1)

#E(Rm) = E(xg) = x*e^(0.0202)
```

```
df_ans["E_Rm"] = df_ans["x"]*np.exp(Rm_coef)

df_ans["ERP"] = df_ans["E_Rm"] - Rf

#parameters setting
xlabel = "b0"
ylabel = "price-divident ratio"
title = "price-dividend ratio vs b0"
##################################################
#plt.figure(figsize=(19,19))
plt.plot(df_ans["b0"], df_ans["price_dividend_ratio"])
plt.xlabel(xlabel)
plt.ylabel(ylabel)
#plt.xticks(df_ans["b0"],rotation=45,fontsize=9)
plt.title(title)
plt.show()

#parameters setting
xlabel = "b0"
ylabel = "equity premium"
title = "equity premium vs b0"
##############################################
plt.xlabel(xlabel)
plt.ylabel(ylabel)
plt.title(title)
plt.plot(df_ans["b0"],df_ans["ERP"])
plt.show()
```



Briefly explain (in words) how to extend the model to incorporate the house money effect.

The house money effect refers to a psychological bias where individuals tend to take greater risks with profits, seen as "house money," as opposed to their initial stake. To incorporate this into a model, you might adjust decision-making parameters based on the perceived gains.

In simple terms, when a person perceives they are working with profits (house money), the model could encourage a more risk-tolerant behavior. This might involve modifying the risk-reward trade-off or adjusting decision thresholds to reflect the psychological tendency to take more risks with perceived gains.

For example, if the model is used in a financial context, it could dynamically adjust risk parameters based on the current profit or loss situation of the individual, encouraging more risk-taking behavior when they are ahead. However, it's essential to strike a balance to avoid excessive risk-taking that could lead to significant losses. The incorporation of the house money effect would involve understanding the user's current financial situation and adapting the decision-making process accordingly.