

```

1 OICD_server <- function(input,output,session) {
2
3 # set to default risky -----
4
5 observeEvent(input$setToDefault_Parameter_x,{
6   updateSliderInput(session, inputId= "capital_x",
7     label="capital",
8     min = 1,
9     max = 100,
10    step = 1,
11    value = 20)
12   updateSliderInput(session, inputId = "price_x",
13     label= "price",
14     min = 1,
15     max = 100,
16     step = 1,
17     value = 19)
18   updateSliderInput(session, inputId = "beta_x",
19     label= "discount factor",
20     min = 0,
21     max = 1,
22     step = 0.01,
23     value = 0.9)
24   updateSliderInput(session, inputId = "gamma_x",
25     label= "constant relative risk aversion",
26     min = 0.01,
27     max = 5,
28     step = 0.01,
29     value = 2)
30   updateSliderInput(session, inputId = "delta_x",
31     label= "isoquant delta",
32     min = -0.8,
33     max = 0.8,
34     step = 0.1,
35     value = 0)
36 })
37
38 observeEvent(input$setToDefault_investment_x,{
39   updateSliderInput(session, inputId = "probability1_x",
40     label= "probability1",
41     min = 0,
42     max = 1,
43     step = 0.01,
44     value = 0.2)
45   updateSliderInput(session, inputId= "dividend1_x",
46     label="dividend1",
47     min = 1,
48     max = 100,
49     step= 1,
50     value = 5)
51   updateSliderInput(session, inputId = "probability2_x",
52     label= "probability2",
53     min = 0,
54     max = 1,
55     step = 0.01,
56     value = 0.5)
57   updateSliderInput(session, inputId= "dividend2_x",
58     label="dividend2",
59     min = 1,
60     max = 100,

```

[illegible]

```

121     else if(input$probability1_x + input$probability3_x < 1){
122         updateSliderInput(session, inputId = "probability2_x",
123                             value = 1 - input$probability1_x - input$probability3_x)}}
124     else if(input$probability1_x + input$probability2_x + input$probability3_x <
125 1){
126         updateSliderInput(session, inputId = "probability2_x",
127                             value = 1 - input$probability1_x - input$probability3_x)}
128     })
129
130
131 # import parameters risky -----
132
133 parameters_x <- reactive({
134     par <- list()
135     par$capital <- input$capital_x
136     par$price <- input$price_x
137     par$beta <- input$beta_x
138     par$gamma <- input$gamma_x
139     par$delta <- input$delta_x
140     return(par)
141 })
142
143 probabilities_x <- reactive({
144     probs <- c(input$probability1_x, input$probability2_x, input$probability3_x)
145     return(probs)
146 })
147
148 dividends_x <- reactive({
149     divs <- c(input$dividend1_x, input$dividend2_x, input$dividend3_x)
150     return(divs)
151 })
152
153 x_axis_x <- reactive({
154     scale_x <- 200
155     return(seq(1/scale_x, (scale_x-1)/scale_x, 1/scale_x)))
156
157
158 # find maximal aggregated utility for given parameters risky -----
159
160 MaxExpTotUtility_x <- reactive({
161     max_x.l <- list()
162     par_x.l <- parameters_x()
163     pi.v <- probabilities_x()
164     d1.v <- dividends_x()
165     x.v <- par_x.l$capital/par_x.l$price*x_axis_x()
166     x.d <- c(min(x.v), max(x.v))
167
168     u.f <- function(c, gamma = par_x.l$gamma){
169         if(gamma == 1){return(log(c))}
170         else{return((c^(1-gamma)-1)/(1-gamma))}}
171
172     bE_u1.f <- function(x, p0 = par_x.l$price, beta = par_x.l$beta, pi = pi.v,
173                         d1 = d1.v){
174         return(beta*(pi[1]*u.f(x*d1[1])+pi[2]*u.f(x*d1[2])+pi[3]*u.f(x*d1[3])))}
175
176     U_x.f <- function(x, K=par_x.l$capital, p0 = par_x.l$price){
177         return(u.f(K - x * p0) + bE_u1.f(x))}
178
179     f.v <- U_x.f(x.v)
180     max_x.l$objective <- max(f.v, na.rm = TRUE)

```

```

181     max_x.l$maximum <- x.v[which.max(f.v)]
182
183     return(max_x.l)
184 })
185
186
187
188 # portfolio choice plot risky -----
189
190 output$PortfolioChoice_x <- renderPlot({
191     max_x.l <- MaxExpTotUtility_x()
192     par_x.l <- parameters_x()
193     pi.v <- probabilities_x()
194     dl.v <- dividends_x()
195
196     x.v <- par_x.l$capital*x_axis_x()
197
198
199     u.f <- function(c, gamma = par_x.l$gamma){
200         if(gamma == 1){return(log(c))}
201         else{return((c^(1-gamma)-1)/(1-gamma))}
202     }
203
204     bE_u1.f <- function(xp0, p0 = par_x.l$price, beta = par_x.l$beta, pi = pi.v,
205                         dl = dl.v){
206         x <- xp0/p0
207         return(beta*(pi[1]*u.f(x*dl[1])+pi[2]*u.f(x*dl[2])+pi[3]*u.f(x*dl[3])))
208     }
209
210     U_x.f <- function(xp0, K=par_x.l$capital, p0=par_x.l$price){
211         return(u.f(K - xp0) + bE_u1.f(xp0))
212     }
213
214     U_x.v <- U_x.f(x.v)
215     u.v <- u.f(par_x.l$capital - x.v)
216     bE_u1.v <- mapply(bE_u1.f, x.v)
217
218
219
220     fmax <- max(u.f(par_x.l$capital - par_x.l$price*max_x.l$maximum),
221                bE_u1.f(par_x.l$price*max_x.l$maximum),
222                U_x.f(par_x.l$price*max_x.l$maximum))
223     fmin <- min(u.f(par_x.l$capital - par_x.l$price*max_x.l$maximum),
224                bE_u1.f(par_x.l$price*max_x.l$maximum),
225                U_x.f(par_x.l$price*max_x.l$maximum))
226
227
228     xmax <- max(x.v)
229     xmin <- min(x.v)
230     ymax <- fmax + (fmax - fmin)
231     ymin <- fmin - (fmax - fmin)
232
233
234     plot(NA,NA,xaxs="r",yaxs="r",xlim=c(xmin,xmax), ylim=c(ymin, ymax),
235          main="Portfolio Choice", xlab="xp0", ylab="U(c0,c1), u(c0), bE[u(c1)]",
236          yaxt='n')
237
238     legend("topright", c("U(xp0)", "u0", "beta*E[u1]"),
239           fill = c("red", "blue", "green"))
240

```

```

241 lines(x.v, U_x.v, col="red")
242 lines(x.v, u.v, col = "blue")
243 lines(x.v, bE_ul.v, col = "green")
244
245 points(par_x.l$price*max_x.l$maximum, max_x.l$objective)
246
247 })
248
249
250
251 # isoquant plot risky -----
252
253 output$Isoquants_x <- renderPlot({
254   max_x.l <- MaxExpTotUtility_x()
255   par_x.l <- parameters_x()
256   pi.v <- probabilities_x()
257   d1.v <- dividends_x()
258
259   x.v <- par_x.l$capital*x_axis_x()
260
261   maximum <- max_x.l$maximum
262   f_max <- max_x.l$objective
263
264   u.f <- function(c, gamma = par_x.l$gamma){
265     if(gamma == 1){return(log(c))}
266     else{return((c^(1-gamma)-1)/(1-gamma))}
267
268   bE_ul.f <- function(x, p0 = par_x.l$price, beta = par_x.l$beta, pi = pi.v,
269     d1 = d1.v){
270     return(beta*(pi[1]*u.f(x/p0*d1[1])+pi[2]*u.f(x/p0*d1[2])+
271       pi[3]*u.f(x/p0*d1[3])))
272
273   U_cx.f <- function(c, xp0, K=par_x.l$capital, p0=par_x.l$price){
274     return(u.f(c) + bE_ul.f(xp0))
275
276   KK <- function(xp0, K = par_x.l$capital){return(K - xp0)}
277
278   cons_x.f <- function(xp0, K=par_x.l$capital, p0=par_x.l$price,
279     beta=par_x.l$beta, gamma=par_x.l$gamma, f_const = max_x.l$objective){
280     if(gamma == 1){c0 <- exp(f_const - bE_ul.f(xp0))}
281     else{c0 <- ((f_const - bE_ul.f(xp0))*(1-gamma)+1)^(1/(1-gamma))}
282     return(c0)}
283
284   xmax <- 1.2*par_x.l$capital
285   xmin <- 0
286   ymax <- 1.2*par_x.l$capital
287   ymin <- 0
288   plot(NA,NA,xaxs="r",yaxs="r",xlim=c(xmin,xmax), ylim=c(ymin, ymax),
289     main="ISOQUANTS", xlab="xp0", ylab="c0")
290
291   lines(x.v, KK(x.v))
292
293
294
295   # Isoquant for U_max
296
297   f_c <- U_cx.f(c = (par_x.l$capital - max_x.l$maximum*par_x.l$price),
298     xp0 = max_x.l$maximum*par_x.l$price, par_x.l$capital)
299   x.v2 <- x.v
300   cons_x.v <- cons_x.f(x.v, f_const=f_c)

```

```

301 cons_x.v[1: which.max(cons_x.v)] <- NA
302 cons_x.v[cons_x.v > par_x.l$capital] <- NA
303
304 k <- which.min(abs(cons_x.v - par_x.l$capital))
305
306 cons_x.v[k - 1] <- par_x.l$capital
307 x.v2[k-1] <- x.v2[k] + (x.v2[k] - x.v2[k+1])/(cons_x.v[k]
308               - cons_x.v[k+1])*(cons_x.v[k-1] - cons_x.v[k])
309 lines(x.v2, cons_x.v)
310
311
312 # Isoquant for U(delta)
313
314 f_c <- U_cx.f(
315   c = (par_x.l$capital-max_x.l$maximum*par_x.l$price)*(1+par_x.l$delta),
316   xp0 = max_x.l$maximum*par_x.l$price*(1+par_x.l$delta))
317 x.v3 <- x.v
318 cons_x.v <- cons_x.f(x.v, f_const=f_c)
319 cons_x.v[1: which.max(cons_x.v)] <- NA
320 cons_x.v[cons_x.v > par_x.l$capital] <- NA
321 k <- which.min(abs(cons_x.v - par_x.l$capital))
322
323 cons_x.v[k - 1] <- par_x.l$capital
324 x.v3[k-1] <- x.v2[k] + (x.v2[k] - x.v2[k+1])/(cons_x.v[k]
325               - cons_x.v[k+1])*(cons_x.v[k-1] - cons_x.v[k])
326 lines(x.v3, cons_x.v)
327
328
329
330
331 points(max_x.l$maximum*par_x.l$price,
332        par_x.l$capital - max_x.l$maximum*par_x.l$price)
333 legend("topright", c(c("xp0* = ", round(max_x.l$maximum*par_x.l$price, 2)),
334                     c("c0* = ",
335                       round(par_x.l$capital - max_x.l$maximum*par_x.l$price, 2))))
336
337 })
338
339
340
341 # marginal utility plot risky -----
342
343 output$MarginalUtility_x <- renderPlot({
344   max_x.l <- MaxExpTotUtility_x()
345   par_x.l <- parameters_x()
346   pi.v <- probabilities_x()
347   d1.v <- dividends_x()
348
349   x.v <- par_x.l$capital*x_axis_x()
350
351
352 u.f <- function(c, gamma = par_x.l$gamma){
353   if(gamma == 1){return(log(c))}
354   else{return((c^(1-gamma)-1)/(1-gamma))}}
355
356 bE_u1.f <- function(x, p0 = par_x.l$price, beta = par_x.l$beta, pi = pi.v,
357                    d1 = d1.v){
358   return(beta*(pi[1]*u.f(x/p0*d1[1])+pi[2]*u.f(x/p0*d1[2])+
359               pi[3]*u.f(x/p0*d1[3]))))
360

```

```

361 U_cx.f <- function(c, xp0, K=par_x.l$capital, p0=par_x.l$price){
362   return(u.f(c) + bE_u1.f(xp0))}
363
364 U_c.f <- function(c, xp0 = max_x.l$maximum*par_x.l$price, K=par_x.l$capital,
365   p0=par_x.l$price){
366   return(u.f(c) + bE_u1.f(xp0))}
367
368 U_x.f <- function(xp0, c = par_x.l$capital-max_x.l$maximum*par_x.l$price,
369   K=par_x.l$capital, p0=par_x.l$price){
370   return(u.f(c) + bE_u1.f(xp0))}
371
372
373 xmax <- par_x.l$capital
374 xmin <- 0
375
376 # fit y_axis limits to the plot
377 p <- max_x.l$maximum*par_x.l$price
378 q <- par_x.l$capital - p
379 x1 <- max(xmin, q - (xmax-xmin)/3, na.rm = TRUE)
380 x2 <- min(xmax, q + (xmax-xmin)/3, na.rm = TRUE)
381 slope <- (U_c.f(q*1.01)-U_c.f(q*0.99))/(q*1.01 - q*0.99)
382 U_c.f1 <- U_c.f(q) + (x1-q)*slope
383 U_c.f2 <- U_c.f(q) + (x2-q)*slope
384 x1 <- max(xmin, p - (xmax-xmin)/3, na.rm = TRUE)
385 x2 <- min(xmax, p + (xmax-xmin)/3, na.rm = TRUE)
386 slope <- (U_x.f(p*1.01)-U_x.f(p*0.99))/(p*1.01 - p*0.99)
387 U_x.f1 <- U_x.f(p) + (x1-p)*slope
388 U_x.f2 <- U_x.f(p) + (x2-p)*slope
389 yaxismax <- max(U_x.f2, U_c.f2)
390 yaxismin <- min(U_x.f1, U_c.f1)
391 yaxisdist <- yaxismax - yaxismin
392
393 ymax <- yaxismax + 0.5*yaxisdist
394 ymin <- yaxismin - 0.5*yaxisdist
395
396 par(mfrow = c(1, 2))
397 plot(NA,NA,xaxs="r",yaxs="r",xlim=c(xmin,xmax), ylim=c(ymin, ymax),
398   main="marginal utility c0", xlab="c0", ylab="", yaxt='n')
399
400 lines(x.v, U_c.f(x.v))
401
402
403 # tangent to immediate consumption part
404 tangent_U_c.f <- function(p){
405   x1 <- max(xmin, p - (xmax-xmin)/3, na.rm = TRUE)
406   x2 <- min(xmax, p + (xmax-xmin)/3, na.rm = TRUE)
407   slope <- (U_c.f(p*1.01)-U_c.f(p*0.99))/(p*1.01 - p*0.99)
408   U_c.f1 <- U_c.f(p) + (x1-p)*slope
409   U_c.f2 <- U_c.f(p) + (x2-p)*slope
410   lines(c(x1, x2), c(U_c.f1, U_c.f2))
411   points(p, U_c.f(p))
412   return(slope)}
413
414 s <- tangent_U_c.f(par_x.l$capital - max_x.l$maximum*par_x.l$price)
415 s <- round(s, abs(log10(s))+1)
416
417 legend("bottomright", c("dU/dc0 = ", s))
418
419
420

```

```

421
422
423
424 plot(NA,NA,xaxs="r",yaxs="r",xlim=c(xmin,xmax),
425       ylim=c(ymin, ymax), main="marginal utility beta*E[u(c1)]",
426       xlab="xp0", ylab="", yaxt='n')
427
428 lines(x.v, U_x.f(x.v))
429
430 # tangent to investment part
431 tangent_U_x.f <- function(p){
432   x1 <- max(xmin, p - (xmax-xmin)/3, na.rm = TRUE)
433   x2 <- min(xmax, p + (xmax-xmin)/3, na.rm = TRUE)
434   slope <- (U_x.f(p*1.01)-U_x.f(p*0.99))/(p*1.01 - p*0.99)
435   U_x.f1 <- U_x.f(p) + (x1-p)*slope
436   U_x.f2 <- U_x.f(p) + (x2-p)*slope
437   lines(c(x1, x2), c(U_x.f1, U_x.f2))
438   points(p, U_x.f(p))
439   return(slope)}
440
441 t <- tangent_U_x.f(max_x.l$maximum*par_x.l$price)
442 t <- round(t, abs(log10(t))+1)
443
444 legend("bottomright", c("dU/dc1 =", t))
445
446
447
448 })
449
450
451
452 # results output risky -----
453
454 output$maxZf_x <- renderText({
455   max_x.l <- MaxExpTotUtility_x()
456   paste("max. exp. Utility: ", round(max_x.l$objective, 3))
457 })
458
459 output$max_x <- renderText({
460   max_x.l <- MaxExpTotUtility_x()
461   paste("x*: ", round(max_x.l$maximum, 3))
462 })
463
464
465
466 # set to default riskless -----
467 -
468
469 observeEvent(input$setToDefault_Parameter_xy,{
470   updateSliderInput(session, inputId= "capital_xy",
471                     label="capital",
472                     min = 1,
473                     max = 100,
474                     step = 1,
475                     value = 20)
476   updateSliderInput(session, inputId = "price_p0_xy",
477                     label= "price risky investment",
478                     min = 1,
479                     max = 100,
480                     step = 1,

```



```

481         value = 19)
482     updateSliderInput(session, inputId = "price_b0_xy",
483         label= "price riskless investment",
484         min = 0.05,
485         max = 5,
486         step = 0.05,
487         value = 0.95)
488     updateSliderInput(session, inputId = "beta_xy",
489         label= "discount factor",
490         min = 0,
491         max = 1,
492         step = 0.01,
493         value = 0.9)
494     updateSliderInput(session, inputId = "gamma_xy",
495         label= "constant relative risk aversion",
496         min = 0.01,
497         max = 5,
498         step = 0.01,
499         value = 2)
500     updateSliderInput(session, inputId = "delta_xy",
501         label= "isoquant delta y",
502         min = -1,
503         max = 1,
504         step = 0.05,
505         value = 0)
506 })
507
508 observeEvent(input$setToDefault_investment_xy,{
509     updateSliderInput(session, inputId = "probability1_xy",
510         label= "probability1",
511         min = 0,
512         max = 1,
513         step = 0.01,
514         value = 0.2)
515     updateSliderInput(session, inputId= "dividend1_xy",
516         label="dividend1",
517         min = 1,
518         max = 100,
519         step= 1,
520         value = 5)
521     updateSliderInput(session, inputId = "probability2_xy",
522         label= "probability2",
523         min = 0,
524         max = 1,
525         step = 0.01,
526         value = 0.5)
527     updateSliderInput(session, inputId= "dividend2_xy",
528         label="dividend2",
529         min = 1,
530         max = 100,
531         step= 1,
532         value = 15)
533
534     updateSliderInput(session, inputId = "probability3_xy",
535         label= "probability3",
536         min = 0,
537         max = 1,
538         step = 0.01,
539         value = 0.3)
540

```

```

541     updateSliderInput(session, inputId= "dividend3_xy",
542                       label="dividend3",
543                       min = 1,
544                       max = 100,
545                       step= 1,
546                       value = 25)
547   })
548
549
550
551 # adjust probabilities riskless -----
552
553 observeEvent(input$probability1_xy,{
554   if(input$probability1_xy+input$probability2_xy+input$probability3_xy > 1){
555     if(input$probability1_xy + input$probability2_xy > 1){
556       updateSliderInput(session, inputId = "probability3_xy", value = 0)
557       updateSliderInput(session, inputId = "probability2_xy",
558                         value = 1 - input$probability1_xy)}
559     else if(input$probability1_xy + input$probability2_xy < 1){
560       updateSliderInput(session, inputId = "probability3_xy",
561                         value = 1 - input$probability1_xy - input$probability2_xy)}}
562   else if(input$probability1_xy+input$probability2_xy+input$probability3_xy <
563 1){
564     updateSliderInput(session, inputId = "probability3_xy",
565                       value = 1 - input$probability1_xy - input$probability2_xy)}
566   })
567
568 observeEvent(input$probability2_xy,{
569   if(input$probability1_xy+input$probability2_xy+input$probability3_xy > 1){
570     if(input$probability1_xy + input$probability2_xy > 1){
571       updateSliderInput(session, inputId = "probability3_xy", value = 0)
572       updateSliderInput(session, inputId = "probability1_xy",
573                         value = 1 - input$probability2_xy)}
574     else if(input$probability1_xy + input$probability2_xy < 1){
575       updateSliderInput(session, inputId = "probability3_xy",
576                         value = 1 - input$probability1_xy - input$probability2_xy)}}
577   else if(input$probability1_xy+input$probability2_xy+input$probability3_xy <
578 1){
579     updateSliderInput(session, inputId = "probability3_xy",
580                       value = 1 - input$probability1_xy - input$probability2_xy)}
581   })
582
583 observeEvent(input$probability3_xy,{
584   if(input$probability1_xy+input$probability2_xy+input$probability3_xy > 1){
585     if(input$probability1_xy + input$probability3_xy > 1){
586       updateSliderInput(session, inputId = "probability2_xy", value = 0)
587       updateSliderInput(session, inputId = "probability1_xy",
588                         value = 1 - input$probability3_xy)}
589     else if(input$probability1_xy + input$probability3_xy < 1){
590       updateSliderInput(session, inputId = "probability2_xy",
591                         value = 1 - input$probability1_xy - input$probability3_xy)}}
592   else if(input$probability1_xy+input$probability2_xy+input$probability3_xy <
593 1){
594     updateSliderInput(session, inputId = "probability2_xy",
595                       value = 1 - input$probability1_xy - input$probability3_xy)}
596   })
597
598
599
600 # import parameters riskless -----

```

```

601
602 parameters_xy <- reactive({
603   par <- list()
604   par$capital <- input$capital_xy
605   par$price <- input$price_p0_xy
606   par$brice <- input$price_b0_xy
607   par$beta <- input$beta_xy
608   par$gamma <- input$gamma_xy
609   par$delta <- input$delta_xy
610   return(par)
611 })
612
613 probabilities_xy <- reactive({
614   probs <- c(input$probability1_xy, input$probability2_xy,
615             input$probability3_xy)
616   return(probs)
617 })
618
619 dividends_xy <- reactive({
620   divs <- c(input$dividend1_xy, input$dividend2_xy, input$dividend3_xy)
621   return(divs)
622 })
623
624 x_axis_xy <- reactive({
625   scale_xy <- 200
626   return(seq(0/scale_xy, (scale_xy - 1)/scale_xy, 1/scale_xy)))
627
628
629
630 # find maximal aggregated utility for given parameters riskless -----
631
632 MaxExpTotUtility_xy <- reactive({
633   max_xy.l <- list()
634   par_xy.l <- parameters_xy()
635   pi.v <- probabilities_xy()
636   d1.v <- dividends_xy()
637   v.v <- par_xy.l$capital*x_axis_xy()
638   v.d <- c(min(v.v), max(v.v))
639
640   u.f <- function(c, gamma = par_xy.l$gamma){
641     if(gamma == 1){return(log(c))}
642     else{return((c^(1-gamma)-1)/(1-gamma))}}
643
644   bE_u1.f <- function(xp0, yb0, p0=par_xy.l$price, b0=par_xy.l$brice, pi=pi.v,
645                      d1=d1.v, beta=par_xy.l$beta){
646     return(beta*(pi[1]*u.f(xp0/p0*d1[1]+yb0/b0))+
647            (pi[2]*u.f(xp0/p0*d1[2]+yb0/b0))+
648            (pi[3]*u.f(xp0/p0*d1[3]+yb0/b0))))
649
650   U_xp0yb0.f <- function(xp0, yb0, K=par_xy.l$capital, p0=par_xy.l$price,
651                          b0=par_xy.l$brice){
652     mapply(function(xp, yb){
653       if(K - xp - yb < 0){return(NA)}
654       else{return(u.f(K - xp - yb) + bE_u1.f(xp, yb))}}, xp = xp0, yb = yb0)}
655
656   f.m <- outer(v.v, v.v, U_xp0yb0.f)
657
658   max_xy.l$objective <- max(f.m, na.rm = TRUE)
659   max_xy.l$maximum_x <- v.v[which.max(f.m)%length(v.v)]
660   max_xy.l$maximum_y <- v.v[which.max(f.m)%length(v.v)+1]

```

```

661
662
663     return(max_xy.l)
664 })
665
666
667
668 # portfolio choice plot riskless -----
669
670 output$PortfolioChoice_xy <- renderPlot({
671     max_xy.l <- MaxExpTotUtility_xy()
672     par_xy.l <- parameters_xy()
673     pi.v <- probabilities_xy()
674     d1.v <- dividends_xy()
675
676     v.v <- par_xy.l$capital*x_axis_xy()
677     v.d <- c(min(v.v), max(v.v))
678
679     u.f <- function(c, gamma = par_xy.l$gamma){
680         if(gamma == 1){return(log(c))}
681         else{return((c^(1-gamma)-1)/(1-gamma))}
682     }
683
684     bE_u1.f <- function(xp0, yb0, p0=par_xy.l$price, b0=par_xy.l$brice, pi=pi.v,
685                         d1=d1.v, beta=par_xy.l$beta){
686         return(beta*(pi[1]*u.f(xp0/p0*d1[1]+yb0/b0))+
687                (pi[2]*u.f(xp0/p0*d1[2]+yb0/b0))+
688                (pi[3]*u.f(xp0/p0*d1[3]+yb0/b0)))
689     }
690
691     U_xp0yb0.f <- function(xp0, yb0, K=par_xy.l$capital, p0=par_xy.l$price,
692                           b0=par_xy.l$brice){
693         mapapply(function(xp, yb){
694             if(K - xp - yb < 0){return(NA)}
695             else{return(u.f(K - xp - yb) + bE_u1.f(xp, yb))}, xp = xp0, yb = yb0)
696         }
697     )
698
699     par(mfrow = c(1, 2))
700
701     U1.v <- U_xp0yb0.f(v.v, max_xy.l$maximum_y)
702
703     if(max_xy.l$maximum_y > 0){
704         U1.v[which.min(U1.v):length(U1.v)] <- NA
705     }
706
707     u1.v <- u.f(par_xy.l$capital - v.v - max_xy.l$maximum_y)
708     u1.v[which.min(u1.v):length(u1.v)] <- NA
709
710     bE_u1.v <- mapapply(bE_u1.f, v.v, max_xy.l$maximum_y)
711
712     fmax <- max(u.f(par_xy.l$capital - max_xy.l$maximum_x - max_xy.l$maximum_y),
713                bE_u1.f(max_xy.l$maximum_x, max_xy.l$maximum_y),
714                U_xp0yb0.f(max_xy.l$maximum_x, max_xy.l$maximum_y))
715     fmin <- min(u.f(par_xy.l$capital - max_xy.l$maximum_x - max_xy.l$maximum_y),
716                bE_u1.f(max_xy.l$maximum_x, max_xy.l$maximum_y),
717                U_xp0yb0.f(max_xy.l$maximum_x, max_xy.l$maximum_y))
718
719     xmax <- max(v.v)
720     xmin <- min(v.v)
721     ymax <- fmax + (fmax - fmin)
722     ymin <- fmin - (fmax - fmin)

```

```

721
722 plot(NA,NA,xaxs="r",yaxs="r",xlim=c(xmin,xmax), ylim=c(ymin, ymax),
723      main=NA, xlab="xp0", ylab=NA, yaxt='n')
724
725 legend("topright", c("U(xp0)", "u0", "beta*E[u1]"),
726      fill = c("red", "blue", "green"))
727
728 lines(v.v, U1.v, col="red")
729 lines(v.v, u1.v,
730      col = "blue")
731 lines(v.v, bE_u1.v, col = "green")
732
733 points(v.v[which.max(U1.v)], max(U1.v, na.rm = TRUE))
734
735
736
737
738
739 U2.v <- U_xp0yb0.f(max_xy.l$maximum_x, v.v)
740
741 if(max_xy.l$maximum_x > 0){
742     U2.v[which.min(U2.v):length(U2.v)] <- NA}
743
744 u2.v <- u.f(par_xy.l$capital - v.v - max_xy.l$maximum_x)
745 u2.v[which.min(u2.v):length(u2.v)] <- NA
746
747 bE_u1.v <- mapply(bE_u1.f, max_xy.l$maximum_x, v.v)
748
749 plot(NA,NA,xaxs="r",yaxs="r",xlim=c(xmin,xmax), ylim=c(ymin, ymax),
750      main=NA, xlab="yb0", ylab="U(c0,c1), u(c0), bE[u(c1)]", yaxt='n')
751
752 legend("topright", c("U(yb0)", "u0", "beta*E[u1]"),
753      fill = c("red", "blue", "green"))
754
755 lines(v.v, U2.v, col="red")
756 lines(v.v, u2.v,
757      col = "blue")
758 lines(v.v, bE_u1.v, col = "green")
759
760 points(v.v[which.max(U2.v)], max(U2.v, na.rm = TRUE))
761
762
763
764 title("PORTFOLIO CHOICE", line = -2, outer=TRUE)
765
766
767
768 })
769
770
771
772 # isoquant plot riskless -----
773
774 output$Isoquants_xy <- renderPlot({
775     max_xy.l <- MaxExpTotUtility_xy()
776     par_xy.l <- parameters_xy()
777     pi.v <- probabilities_xy()
778     d1.v <- dividends_xy()
779
780     v.v <- par_xy.l$capital*x_axis_xy()

```

```

781 v.d <- c(min(v.v), max(v.v))
782
783 u.f <- function(c, gamma = par_xy.l$gamma){
784   if(gamma == 1){return(log(c))}
785   else{return((c^(1-gamma)-1)/(1-gamma))}}
786
787 bE_ul.f <- function(xp0, yb0, p0=par_xy.l$price, b0=par_xy.l$brice, pi=pi.v,
788   d1=d1.v, beta=par_xy.l$beta){
789   return(beta*(pi[1]*u.f(xp0/p0*d1[1]+yb0/b0))+
790     (pi[2]*u.f(xp0/p0*d1[2]+yb0/b0))+
791     (pi[3]*u.f(xp0/p0*d1[3]+yb0/b0)))}
792
793 U_c0xp0.f <- function(xp0, yb0=max_xy.l$maximum_y, K=par_xy.l$capital,
794   p0=par_xy.l$price, b0=par_xy.l$brice){
795   mapply(function(xp, yb){
796     if(K - xp - yb < 0){return(NA)}
797     else{return(u.f(K - xp - yb) + bE_ul.f(xp, yb))}},
798     xp = xp0, yb = yb0)}
799
800 KK <- function(xp0, K=par_xy.l$capital, yb0=max_xy.l$maximum_y){
801   return(K - xp0 - yb0)}
802
803 cons_f.f <- function(xp0, yb0=max_xy.l$maximum_y, K=par_xy.l$capital,
804   p0=par_xy.l$price, b0=par_xy.l$brice,
805   beta=par_xy.l$beta, gamma = par_xy.l$gamma,
806   f_const=max_xy.l$objective){
807   if(gamma == 1){
808     c0 <- exp(f_const - bE_ul.f(xp0, yb0))}
809   else{c0 <- ((f_const - bE_ul.f(xp0, yb0))*(1-gamma)+1)^(1/(1-gamma))
810     }
811   return(c0)}
812
813
814 xmax <- 1.2*par_xy.l$capital
815 xmin <- 0
816 ymax <- 1.2*par_xy.l$capital
817 ymin <- 0
818
819 plot(NA, NA, xaxs="r", yaxs="r", xlim=c(xmin, xmax), ylim=c(ymin,ymax),
820   main="ISOQUANTS", xlab="xp0", ylab="c0")
821
822 vK.v <- head(v.v, which.min(abs(KK(v.v))))
823 lines(vK.v, KK(vK.v))
824
825 v.v2 <- v.v
826 cons_f.v <- cons_f.f(v.v2)
827 cons_f.v[1: which.max(cons_f.v)] <- NA
828 cons_f.v[cons_f.v > par_xy.l$capital] <- NA
829
830 k <- which.min(abs(cons_f.v - par_xy.l$capital))
831 cons_f.v[k - 1] <- par_xy.l$capital
832 v.v2[k-1] <- v.v2[k] + (v.v2[k] - v.v2[k+1])/(cons_f.v[k]-cons_f.v[k+1])*
833   (cons_f.v[k-1]-cons_f.v[k])
834
835 lines(v.v2, cons_f.v)
836
837
838
839
840

```

```

841
842 delta_yb0 <- max_xy.l$maximum_y * par_xy.l$delta
843 yb0_new <- max_xy.l$maximum_y * (1 + par_xy.l$delta)
844
845 KK.v <- KK(v.v, K= par_xy.l$capital, yb0 = yb0_new)
846 KK.v[KK.v < 0] <- NA
847 lines(v.v, KK.v, lty = "dashed")
848
849 U_c0xp0.v <- U_c0xp0.f(v.v, yb0 = yb0_new)
850 f_new <- max(U_c0xp0.v, na.rm = TRUE)
851 xp0_new <- which.max(U_c0xp0.v)
852
853 v.v3 <- v.v
854 cons_f2.v <- cons_f.f(v.v, yb0=yb0_new, f_const=f_new)
855
856 cons_f2.v[1: which.max(cons_f2.v)] <- NA
857 cons_f2.v[cons_f2.v > par_xy.l$capital] <- NA
858
859 n <- which.min(abs(cons_f2.v - par_xy.l$capital))
860 cons_f2.v[n - 1] <- par_xy.l$capital
861 v.v3[n-1] <- v.v3[n] + (v.v3[n] - v.v3[n+1])/(cons_f2.v[n]-cons_f2.v[n+1]) *
862   (cons_f2.v[n-1]-cons_f2.v[n])
863
864 lines(v.v3, cons_f2.v, lty = "dashed")
865
866 points(max_xy.l$maximum_x,
867        par_xy.l$capital - max_xy.l$maximum_x - max_xy.l$maximum_y)
868
869 legend("topright", c(c("x*p0 =", "c0* =", "", "y*b0 =", "yb0_new =", "",
870                        "U* =", "U*_new ="),
871                      c(round(max_xy.l$maximum_x, log10(max_xy.l$maximum_x)+3),
872                        round(par_xy.l$capital - max_xy.l$maximum_x -max_xy.l$maximum_y,
873                              log10(par_xy.l$capital - max_xy.l$maximum_x -max_xy.l$maximum_y)+3),
874                        "", round(max_xy.l$maximum_y, log10(max_xy.l$maximum_y)+3),
875                        round(yb0_new, log10(abs(yb0_new))+3), "",
876                        round(max_xy.l$objective, log10(max_xy.l$objective)+3),
877                        round(f_new, log10(f_new)+3))),
878        ncol = 2)
879 })
880
881
882 # marginal utility plot riskless -----
883
884 output$MarginalUtility_xy <- renderPlot({
885   max_xy.l <- MaxExpTotUtility_xy()
886   par_xy.l <- parameters_xy()
887   pi.v <- probabilities_xy()
888   d1.v <- dividends_xy()
889
890   v.v <- par_xy.l$capital*x_axis_xy()
891   v.d <- c(min(v.v), max(v.v))
892
893   u.f <- function(c, gamma = par_xy.l$gamma){
894     if(gamma == 1){return(log(c))}
895     else{return((c^(1-gamma)-1)/(1-gamma))}
896   }
897
898   bE_u1.f <- function(xp0, yb0, p0=par_xy.l$price, b0=par_xy.l$brice, pi=pi.v,
899                      d1=d1.v, beta=par_xy.l$beta){
900     return(beta*(pi[1]*u.f(xp0/p0*d1[1]+yb0/b0))+
901            (pi[2]*u.f(xp0/p0*d1[2]+yb0/b0))+

```

```

901         (pi[3]*u.f(xp0/p0*d1[3]+yb0/b0)))}
902
903 U_c0.f <- function(c0, xp0=max_xy.l$maximum_x, yb0=max_xy.l$maximum_y,
904                 K=par_xy.l$capital, p0=par_xy.l$price, b0=par_xy.l$brice)
905     {return(u.f(c0) + bE_u1.f(xp0, yb0))}
906
907 U_xp0.f <- function(xp0, yb0=max_xy.l$maximum_y, K=par_xy.l$capital,
908                 p0=par_xy.l$price, b0=par_xy.l$brice)
909     {return(u.f(K - max_xy.l$maximum_x - yb0) + bE_u1.f(xp0, yb0))}
910
911 U_yb0.f <- function(yb0, xp0=max_xy.l$maximum_x, K=par_xy.l$capital,
912                 p0=par_xy.l$price, b0=par_xy.l$brice)
913     {return(u.f(K - max_xy.l$maximum_y - xp0) + bE_u1.f(xp0, yb0))}
914
915 xmax <- par_xy.l$capital
916 xmin <- 0
917
918 # fit y_axis limits to the plot
919 p <- par_xy.l$capital - max_xy.l$maximum_x - max_xy.l$maximum_y
920
921 p1 <- max(xmin, p - (xmax-xmin)/3, na.rm = TRUE)
922 p2 <- min(xmax, p + (xmax-xmin)/3, na.rm = TRUE)
923 slope <- (U_c0.f(p+0.01*xmax)-U_c0.f(p-0.01*xmax))/(0.02*xmax)
924 yaxismax <- U_c0.f(p) + (p2-p)*slope
925 yaxismin <- U_c0.f(p) + (p1-p)*slope
926 yaxisdist <- yaxismax - yaxismin
927 ymax <- yaxismax + yaxisdist
928 ymin <- yaxismin - yaxisdist
929
930 par(mfrow = c(1, 3))
931 plot(NA,NA,xaxs="r",yaxs="r",xlim=c(xmin, xmax), ylim=c(ymin, ymax),
932      main="marginal utility c0", xlab="c0", ylab="", yaxt='n')
933
934 lines(v.v, U_c0.f(v.v))
935
936 # tangent to immediate consumption part
937 tangent_U_c0.f <- function(p){
938     x1 <- max(xmin, p - (xmax-xmin)/3, na.rm = TRUE)
939     x2 <- min(xmax, p + (xmax-xmin)/3, na.rm = TRUE)
940     slope <- (U_c0.f(p+0.01*xmax)-U_c0.f(p-0.01*xmax))/(0.02*xmax)
941     U_c0.f1 <- U_c0.f(p) + (x1-p)*slope
942     U_c0.f2 <- U_c0.f(p) + (x2-p)*slope
943     lines(c(x1, x2), c(U_c0.f1, U_c0.f2))
944     points(p, U_c0.f(p))
945     return(slope)}
946
947 s <- tangent_U_c0.f(par_xy.l$capital-max_xy.l$maximum_x-max_xy.l$maximum_y)
948 s <- round(s, abs(log10(s))+1)
949
950 legend("bottomright", c("dU/dc0 = ", s))
951
952
953
954 plot(NA,NA,xaxs="r",yaxs="r",xlim=c(xmin, xmax), ylim=c(ymin, ymax),
955      main="marginal utility beta*E[u(xp0)]", xlab="xp0", ylab="", yaxt='n')
956
957 lines(v.v, U_xp0.f(v.v))
958
959 # tangent to risky investment part
960 tangent_U_xp0.f <- function(p){

```



```

961     x1 <- max(xmin, p - (xmax-xmin)/3, na.rm = TRUE)
962     x2 <- min(xmax, p + (xmax-xmin)/3, na.rm = TRUE)
963     slope <- (U_xp0.f(p+0.01*xmax)-U_xp0.f(p-0.01*xmax))/(0.02*xmax)
964     U_xp0.f1 <- U_xp0.f(p) + (x1-p)*slope
965     U_xp0.f2 <- U_xp0.f(p) + (x2-p)*slope
966     lines(c(x1, x2), c(U_xp0.f1, U_xp0.f2))
967     points(p, U_xp0.f(p))
968     return(slope)}
969
970 t <- tangent_U_xp0.f(max_xy.l$maximum_x)
971 t <- round(t, abs(log10(t))+1)
972
973 legend("bottomright", c("dU/dxp0 = ", t))
974
975
976 plot(NA,NA,xaxs="r",yaxs="r",xlim=c(xmin, xmax), ylim=c(ymin, ymax),
977      main="marginal utility beta*E[u(yb0)]", xlab="yb0", ylab="", yaxt='n')
978
979 lines(v.v, U_yb0.f(v.v))
980
981 # tangent to riskless investment part
982 tangent_U_yb0.f <- function(p){
983     x1 <- max(xmin, p - (xmax-xmin)/3, na.rm = TRUE)
984     x2 <- min(xmax, p + (xmax-xmin)/3, na.rm = TRUE)
985     slope <- (U_yb0.f(p+0.01*xmax)-U_yb0.f(p-0.01*xmax))/(0.02*xmax)
986     U_yb0.f1 <- U_yb0.f(p) + (x1-p)*slope
987     U_yb0.f2 <- U_yb0.f(p) + (x2-p)*slope
988     lines(c(x1, x2), c(U_yb0.f1, U_yb0.f2))
989     points(p, U_yb0.f(p))
990     return(slope)}
991
992 u <- tangent_U_yb0.f(max_xy.l$maximum_y)
993 u <- round(u, abs(log10(u))+1)
994
995 legend("bottomright", c("dU/dyb0 = ", u))
996 })
997
998
999
1000 # results output riskless -----
1001
1002 output$maxZf_xy <- renderText({
1003     max_xy.l <- MaxExpTotUtility_xy()
1004     paste("max. exp. Utility: ", round(max_xy.l$objective, 3))
1005 })
1006
1007 output$max_xy_x <- renderText({
1008     max_xy.l <- MaxExpTotUtility_xy()
1009     paste("maximum in x*p0: ", round(max_xy.l$maximum_x, 3))
1010 })
1011
1012 output$max_xy_y <- renderText({
1013     max_xy.l <- MaxExpTotUtility_xy()
1014     paste("maximum in y*b0: ", round(max_xy.l$maximum_y, 3))
1015 })
1016 }

```