In [1]:

```
#convert txt to csv
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

```
dfg = pd.read_csv('wat-good-time-interval-10000.csv')
```

In [3]:

```
dfm = pd.read_csv('wat-mal-time-interval-10000.csv')
```

In [4]:

```
dfg.head()
```

Out[4]:

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_north	inport_south	i
0	10000	613	194	105	48	385	110	105	70	264	...	695	711	
1	20000	629	266	202	89	337	122	129	50	175	...	544	700	
2	30000	672	266	172	86	316	108	80	65	200	...	553	586	
3	40000	697	290	196	109	341	115	72	84	236	...	614	639	
4	50000	558	243	152	79	249	102	91	67	165	...	397	436	

5 rows × 32 columns

In [5]:

```
dfm.head()
```

Out[5]:

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_north	inport_south
0	10000	675	825	201	95	445	521	179	117	300	...	1479	1076
1	20000	707	860	384	171	419	406	182	76	207	...	1072	1085
2	30000	776	1015	467	235	375	437	205	169	232	...	1148	1033
3	40000	782	1027	409	217	418	465	190	142	287	...	1207	1015
4	50000	690	941	445	242	367	378	198	151	184	...	754	827

5 rows × 32 columns

In [6]:

```
df = dfg.append(dfm, ignore_index=True)
```

In [7]:

```
df.head()
```

Out[7]:

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_north	inport_south	i
0	10000	613	194	105	48	385	110	105	70	264	...	695	711	
1	20000	629	266	202	89	337	122	129	50	175	...	544	700	
2	30000	672	266	172	86	316	108	80	65	200	...	553	586	
3	40000	697	290	196	109	341	115	72	84	236	...	614	639	
4	50000	558	243	152	79	249	102	91	67	165	...	397	436	

5 rows × 32 columns

In [8]:

```
df = df.sort_values('time_intervals')
```

In [9]:

```
df.head(50)
```

Out[9]:

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_north	inport_sou
0	10000	613	194	105	48	385	110	105	70	264	...	695	7
315	10000	675	825	201	95	445	521	179	117	300	...	1479	10
316	20000	707	860	384	171	419	406	182	76	207	...	1072	10
1	20000	629	266	202	89	337	122	129	50	175	...	544	7(
2	30000	672	266	172	86	316	108	80	65	200	...	553	5
317	30000	776	1015	467	235	375	437	205	169	232	...	1148	10
3	40000	697	290	196	109	341	115	72	84	236	...	614	6
318	40000	782	1027	409	217	418	465	190	142	287	...	1207	10
4	50000	558	243	152	79	249	102	91	67	165	...	397	4
319	50000	690	941	445	242	367	378	198	151	184	...	754	8
320	60000	851	1082	532	258	511	441	221	137	295	...	1170	10
5	60000	767	350	298	164	411	125	161	115	237	...	612	6
321	70000	822	709	345	188	463	369	215	224	289	...	908	8
6	70000	762	326	242	139	388	150	139	163	226	...	578	6
322	80000	606	330	234	120	264	60	120	84	180	...	462	5
7	80000	555	301	252	126	227	31	120	78	185	...	459	5(
323	90000	534	300	210	120	210	60	114	102	132	...	384	4
8	90000	507	240	204	120	246	66	114	114	156	...	462	5(
324	100000	528	264	210	132	216	24	96	102	150	...	426	5
9	100000	505	216	180	96	253	30	102	102	174	...	480	4(
325	110000	738	325	247	132	371	126	115	54	221	...	558	5
10	110000	705	334	261	141	335	144	126	56	168	...	441	5
326	120000	704	350	258	156	341	163	132	126	179	...	474	5
11	120000	746	327	219	129	407	169	138	154	251	...	612	5
12	130000	493	317	276	132	180	48	60	42	162	...	330	4
327	130000	562	287	234	126	276	72	84	84	222	...	468	5(
13	140000	413	241	210	96	156	30	30	60	156	...	330	4
328	140000	427	294	258	126	121	36	36	54	121	...	284	4(
329	150000	393	204	180	126	173	18	36	54	167	...	340	3
14	150000	480	276	228	144	144	24	36	54	120	...	264	4
330	160000	224	138	96	60	42	12	6	18	24	...	54	1(
15	160000	60	42	24	12	18	12	0	0	0	...	12	
16	170000	84	48	48	36	36	12	24	18	24	...	54	
331	170000	84	48	42	30	36	24	24	12	6	...	36	

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_north	inport_sou
17	180000	60	30	12	6	18	6	6	6	18	...	54	
332	180000	60	24	18	12	36	6	6	12	36	...	84	
333	190000	84	48	30	24	24	18	12	0	0	...	18	
18	190000	84	42	36	30	42	24	18	12	12	...	36	
334	200000	60	24	18	12	36	12	12	18	30	...	54	
19	200000	66	43	37	30	23	6	7	6	23	...	51	
20	210000	60	42	30	18	6	0	0	6	6	...	18	
335	210000	84	60	48	30	12	0	6	6	12	...	36	
336	220000	84	48	36	36	36	18	0	0	18	...	36	
21	220000	84	42	30	24	42	18	6	0	24	...	48	
22	230000	60	24	18	12	36	6	6	12	36	...	84	
337	230000	60	24	18	6	36	6	12	6	36	...	78	
23	240000	84	60	54	36	24	18	12	0	0	...	18	
338	240000	414	408	408	120	6	0	0	6	6	...	18	
24	250000	84	42	30	30	42	18	6	12	36	...	66	
339	250000	150	120	114	108	30	24	18	12	0	...	24	

50 rows × 32 columns

In [10]:

```
df['tot_packets']= df.iloc[:, 1:16].sum(axis=1)
```

In [11]:

```
df['tot_mean']= df.iloc[:, 1:16].mean(axis=1)
```

In [12]:

```
df
```

Out[12]:

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_east	inport_west
0	10000	613	194	105	48	385	110	105	70	264	...	179	200
315	10000	675	825	201	95	445	521	179	117	300	...	351	393
316	20000	707	860	384	171	419	406	182	76	207	...	453	548
1	20000	629	266	202	89	337	122	129	50	175	...	232	327
2	30000	672	266	172	86	316	108	80	65	200	...	293	362
...	...	...	...	...	...	...	...	...	...	...	...	...	...
312	3130000	559	229	168	102	329	90	96	72	239	...	498	403
628	3140000	486	282	228	120	192	66	102	84	109	...	312	474
313	3140000	503	269	198	114	210	72	90	96	126	...	324	449
314	3150000	747	369	288	198	366	134	114	90	204	...	496	621
629	3150000	762	348	258	180	390	144	102	90	227	...	510	570

630 rows × 34 columns

In [13]:

```
df.to_csv('wat-time-interval-10000.csv',index=False)
```

In [14]:

```
df = pd.read_csv('wat-time-interval-10000.csv')
```

In [15]:

```
df.head(50)
```

Out[15]:

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_east	inport_west
0	10000	613	194	105	48	385	110	105	70	264	...	179	200
1	10000	675	825	201	95	445	521	179	117	300	...	351	393
2	20000	707	860	384	171	419	406	182	76	207	...	453	548
3	20000	629	266	202	89	337	122	129	50	175	...	232	327
4	30000	672	266	172	86	316	108	80	65	200	...	293	362
5	30000	776	1015	467	235	375	437	205	169	232	...	723	778
6	40000	697	290	196	109	341	115	72	84	236	...	385	414
7	40000	782	1027	409	217	418	465	190	142	287	...	806	747
8	50000	558	243	152	79	249	102	91	67	165	...	350	378
9	50000	690	941	445	242	367	378	198	151	184	...	654	746
10	60000	851	1082	532	258	511	441	221	137	295	...	1024	929
11	60000	767	350	298	164	411	125	161	115	237	...	552	616
12	70000	822	709	345	188	463	369	215	224	289	...	827	809
13	70000	762	326	242	139	388	150	139	163	226	...	526	607
14	80000	606	330	234	120	264	60	120	84	180	...	450	552
15	80000	555	301	252	126	227	31	120	78	185	...	447	547
16	90000	534	300	210	120	210	60	114	102	132	...	402	510
17	90000	507	240	204	120	246	66	114	114	156	...	462	494
18	100000	528	264	210	132	216	24	96	102	150	...	426	510
19	100000	505	216	180	96	253	30	102	102	174	...	481	462
20	110000	738	325	247	132	371	126	115	54	221	...	522	536
21	110000	705	334	261	141	335	144	126	56	168	...	430	538
22	120000	704	350	258	156	341	163	132	126	179	...	444	590
23	120000	746	327	219	129	407	169	138	154	251	...	604	597
24	130000	493	317	276	132	180	48	60	42	162	...	312	480
25	130000	562	287	234	126	276	72	84	84	222	...	468	503
26	140000	413	241	210	96	156	30	30	60	156	...	330	408
27	140000	427	294	258	126	121	36	36	54	121	...	294	474
28	150000	393	204	180	126	173	18	36	54	167	...	330	372
29	150000	480	276	228	144	144	24	36	54	120	...	264	426
30	160000	224	138	96	60	42	12	6	18	24	...	54	168
31	160000	60	42	24	12	18	12	0	0	0	...	12	42
32	170000	84	48	48	36	36	12	24	18	24	...	54	96
33	170000	84	48	42	30	36	24	24	12	6	...	36	84

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_east	inport_west
34	180000	60	30	12	6	18	6	6	6	18	...	54	42
35	180000	60	24	18	12	36	6	6	12	36	...	84	54
36	190000	84	48	30	24	24	18	12	0	0	...	18	60
37	190000	84	42	36	30	42	24	18	12	12	...	36	72
38	200000	60	24	18	12	36	12	12	18	30	...	54	54
39	200000	66	43	37	30	23	6	7	6	23	...	39	74
40	210000	60	42	30	18	6	0	0	6	6	...	18	54
41	210000	84	60	48	30	12	0	6	6	12	...	36	84
42	220000	84	48	36	36	36	18	0	0	18	...	36	66
43	220000	84	42	30	24	42	18	6	0	24	...	48	60
44	230000	60	24	18	12	36	6	6	12	36	...	84	54
45	230000	60	24	18	6	36	6	12	6	36	...	78	48
46	240000	84	60	54	36	24	18	12	0	0	...	18	84
47	240000	414	408	408	120	6	0	0	6	6	...	18	486
48	250000	84	42	30	30	42	18	6	12	36	...	66	66
49	250000	150	120	114	108	30	24	18	12	0	...	24	180

50 rows × 34 columns

In [16]:

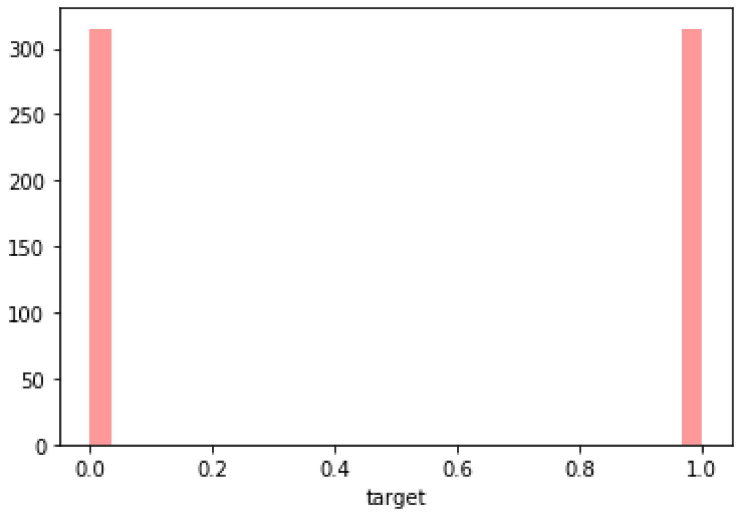
```
import seaborn as sns
```

In [17]:

```
sns.distplot(df['target'], kde = False, bins=30, color='red')
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d25f0dd4c8>
```



In [18]:

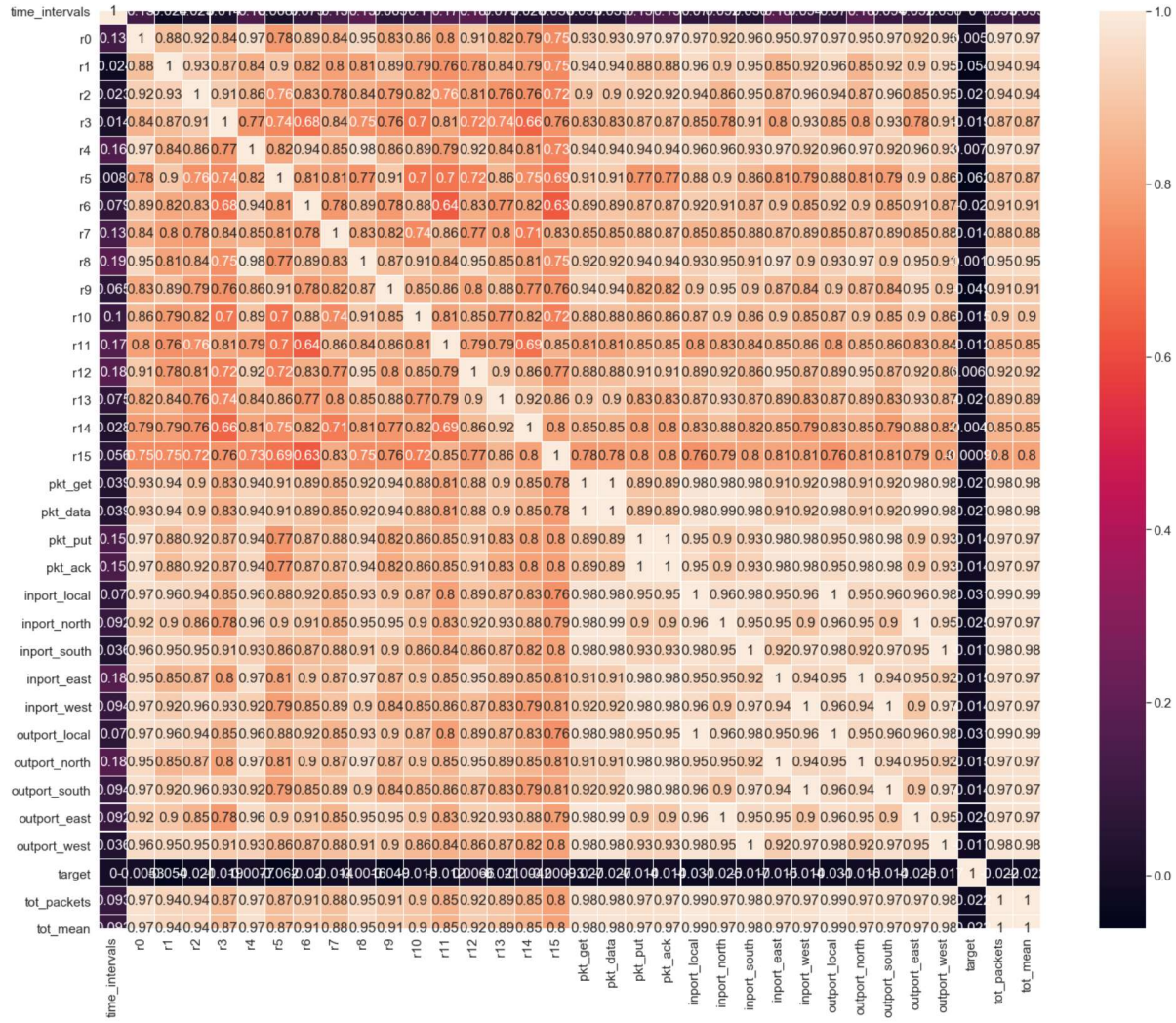
```
df.isnull().sum()
```

Out[18]:

```
time_intervals    0
r0                0
r1                0
r2                0
r3                0
r4                0
r5                0
r6                0
r7                0
r8                0
r9                0
r10               0
r11               0
r12               0
r13               0
r14               0
r15               0
pkt_get           0
pkt_data          0
pkt_put           0
pkt_ack           0
inport_local      0
inport_north      0
inport_south      0
inport_east       0
inport_west       0
outport_local     0
outport_north     0
outport_south     0
outport_east      0
outport_west      0
target            0
tot_packets       0
tot_mean          0
dtype: int64
```

In [19]:

```
plt.figure(figsize=(25,20))
plt.rcParams["axes.labelsize"] = 20
sns.set(font_scale=1.4)
sns.heatmap(df.corr(), annot = True ,linewidths=.1)
plt.show()
```



In [20]:

```
df_corr = df.corr()
df_corr
```

Out[20]:

	time_intervals	r0	r1	r2	r3	r4	r5	
time_intervals	1.000000	0.126189	-0.024202	0.023317	0.014263	0.156730	0.008746	0.(
r0	0.126189	1.000000	0.880315	0.921483	0.843099	0.967894	0.779053	0.8
r1	-0.024202	0.880315	1.000000	0.928026	0.874358	0.838893	0.902357	0.8
r2	0.023317	0.921483	0.928026	1.000000	0.914807	0.862259	0.758288	0.8
r3	0.014263	0.843099	0.874358	0.914807	1.000000	0.772269	0.739808	0.(
r4	0.156730	0.967894	0.838893	0.862259	0.772269	1.000000	0.819686	0.9
r5	0.008746	0.779053	0.902357	0.758288	0.739808	0.819686	1.000000	0.8
r6	0.078534	0.893604	0.820267	0.834735	0.683599	0.936351	0.807417	1.(
r7	0.134347	0.843027	0.800133	0.780864	0.837076	0.847388	0.811944	0.7
r8	0.188715	0.948662	0.806011	0.838658	0.754123	0.979600	0.765015	0.8
r9	0.065188	0.827457	0.886674	0.790391	0.764939	0.859911	0.908614	0.7
r10	0.102202	0.860886	0.787954	0.820909	0.698767	0.892640	0.695586	0.8
r11	0.172795	0.796771	0.763521	0.756715	0.814951	0.788356	0.696688	0.(
r12	0.179868	0.908541	0.776521	0.806334	0.721701	0.924290	0.718594	0.8
r13	0.074840	0.822733	0.843763	0.764902	0.738184	0.838230	0.855968	0.7
r14	0.028335	0.787450	0.791578	0.759947	0.658551	0.809367	0.751011	0.8
r15	0.056096	0.746831	0.745869	0.718016	0.763804	0.726475	0.691399	0.(
pkt_get	0.038868	0.928516	0.944195	0.898233	0.833638	0.937455	0.912105	0.8
pkt_data	0.038957	0.929048	0.944010	0.898550	0.833667	0.938177	0.911528	0.8
pkt_put	0.151557	0.970968	0.875951	0.924178	0.870777	0.943584	0.772211	0.
pkt_ack	0.151576	0.971068	0.875429	0.924264	0.870371	0.943806	0.770952	0.8
inport_local	0.070300	0.970680	0.956778	0.936481	0.853645	0.957091	0.883817	0.9
inport_north	0.092394	0.920258	0.901315	0.855046	0.781116	0.957884	0.901952	0.9
inport_south	0.036279	0.955375	0.949886	0.950114	0.911695	0.927164	0.862797	0.8
inport_east	0.176981	0.951722	0.851538	0.870375	0.803435	0.968424	0.807444	0.8
inport_west	0.093501	0.968819	0.916082	0.960831	0.930476	0.919094	0.791908	0.8
outport_local	0.070297	0.970715	0.956777	0.936627	0.853811	0.957128	0.883655	0.9
outport_north	0.177002	0.951749	0.851558	0.870393	0.803472	0.968449	0.807479	0.8
outport_south	0.093503	0.968830	0.916106	0.960823	0.930422	0.919087	0.791940	0.8
outport_east	0.092382	0.920094	0.901163	0.854758	0.780908	0.957746	0.901946	0.9
outport_west	0.036273	0.955349	0.949885	0.950103	0.911616	0.927126	0.862834	0.8
target	0.000000	-0.005309	-0.053992	-0.020834	-0.018737	-0.007652	-0.061772	-0.(
tot_packets	0.093126	0.974987	0.938291	0.935989	0.872597	0.967581	0.871561	0.9
tot_mean	0.093126	0.974987	0.938291	0.935989	0.872597	0.967581	0.871561	0.9

34 rows × 34 columns

In [21]:

```
df_corr['target']
```

Out[21]:

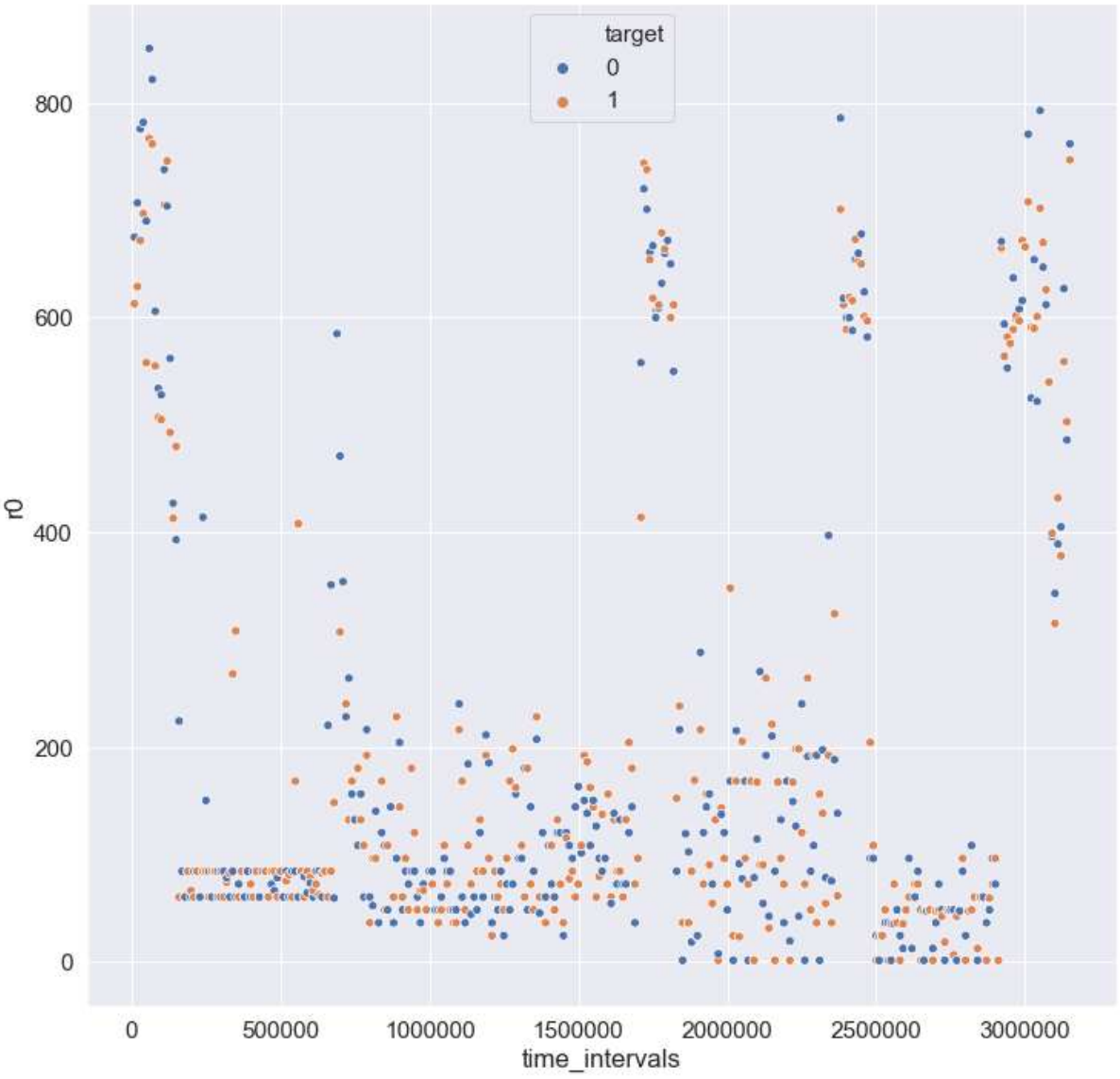
```
time_intervals    0.000000
r0               -0.005309
r1               -0.053992
r2               -0.020834
r3               -0.018737
r4               -0.007652
r5               -0.061772
r6               -0.019972
r7               -0.014347
r8               -0.001649
r9               -0.049409
r10              -0.015207
r11              -0.012263
r12               0.006575
r13              -0.021119
r14              -0.004152
r15              -0.000925
pkt_get          -0.026866
pkt_data         -0.026882
pkt_put          -0.014233
pkt_ack          -0.014278
inport_local     -0.030686
inport_north     -0.025127
inport_south     -0.017061
inport_east      -0.015180
inport_west      -0.014465
outport_local    -0.030690
outport_north    -0.015172
outport_south    -0.014466
outport_east     -0.025125
outport_west     -0.017060
target            1.000000
tot_packets      -0.022066
tot_mean         -0.022066
Name: target, dtype: float64
```

In [22]:

```
dff = pd.read_csv('wat-time-interval-10000.csv')
```

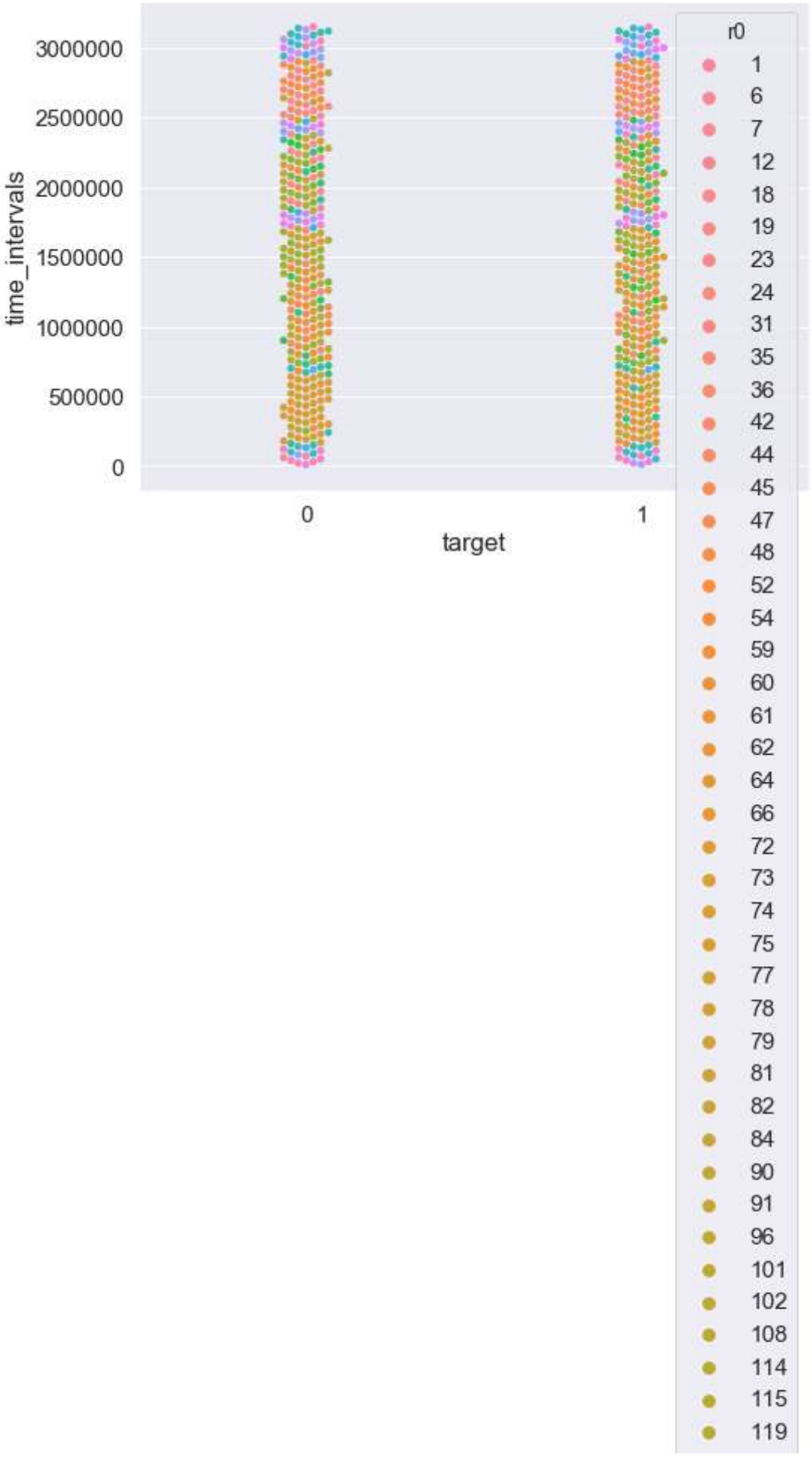
In [23]:

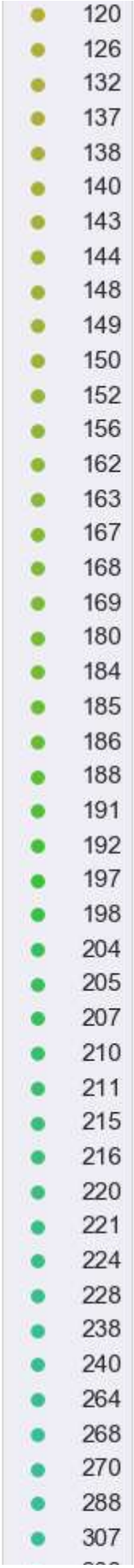
```
plt.figure(figsize=(12,12))
sns.scatterplot(x='time_intervals',y='r0',data=dff, hue='target')
plt.show()
```

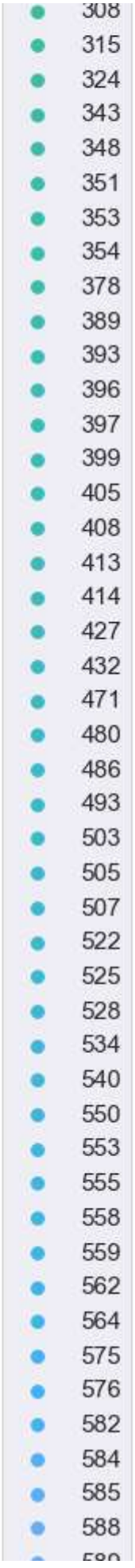


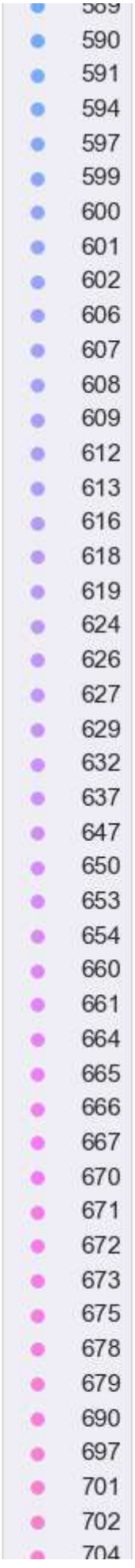
In [24]:

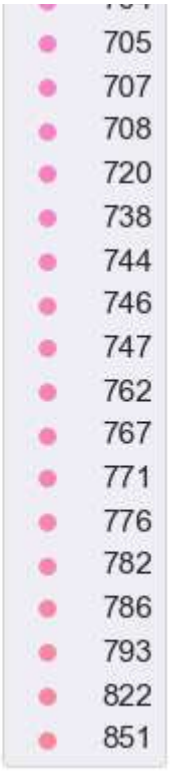
```
plt.figure(figsize=(8,6))
sns.swarmplot(x='target',y='time_intervals',data=dff, hue='r0')
plt.show()
```


time_intervals
3000000
2500000
2000000
1500000
1000000
500000
0
0
1
target
r0
1
6
7
12
18
19
23
24
31
35
36
42
44
45
47
48
52
54
59
60
61
62
64
66
72
73
74
75
77
78
79
81
82
84
90
91
96
101
102
108
114
115
119


120
126
132
137
138
140
143
144
148
149
150
152
156
162
163
167
168
169
180
184
185
186
188
191
192
197
198
204
205
207
210
211
215
216
220
221
224
228
238
240
264
268
270
288
307

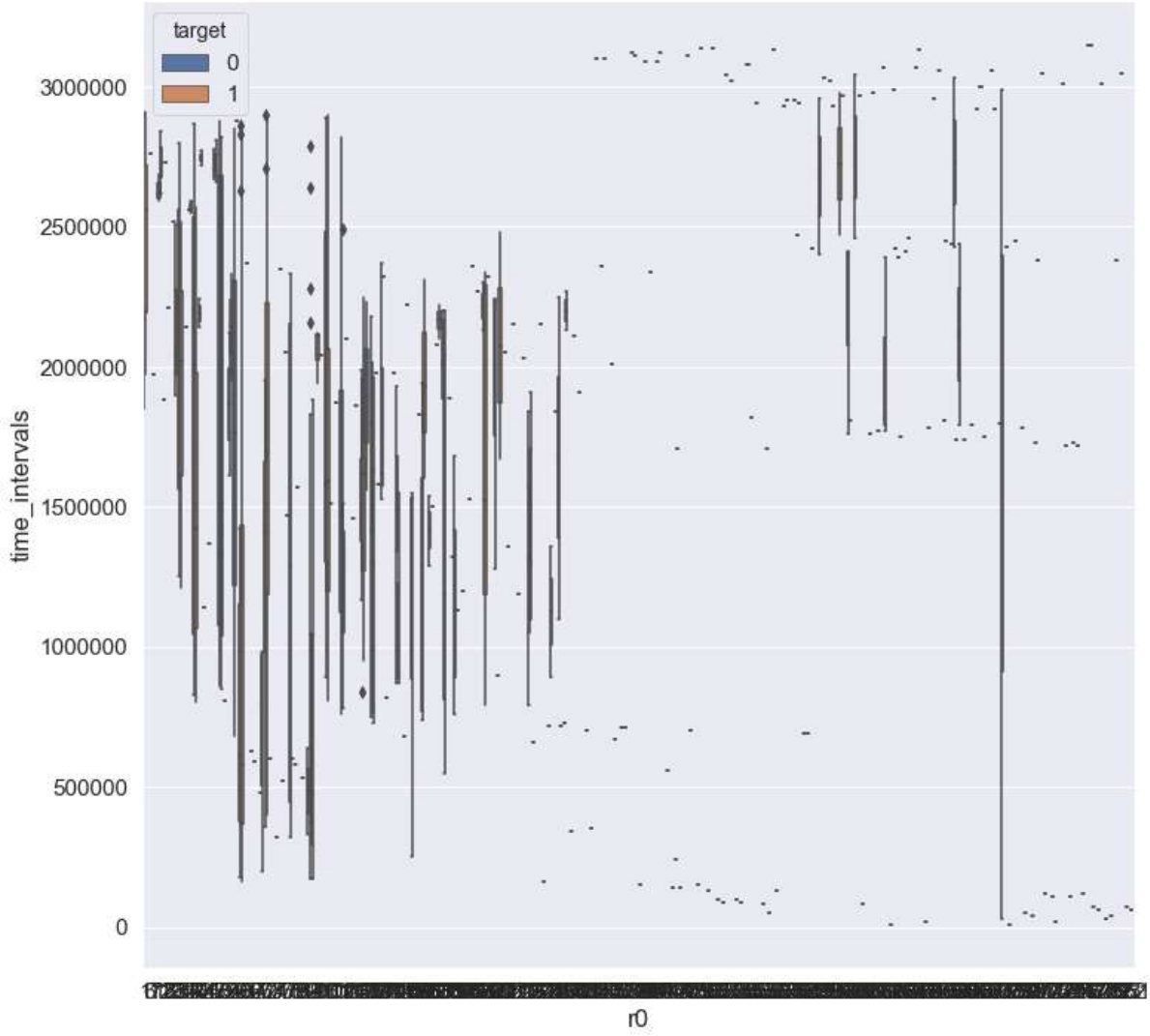

308
315
324
343
348
351
353
354
378
389
393
396
397
399
405
408
413
414
427
432
471
480
486
493
503
505
507
522
525
528
534
540
550
553
555
558
559
562
564
575
576
582
584
585
588


589
590
591
594
597
599
600
601
602
606
607
608
609
612
613
616
618
619
624
626
627
629
632
637
647
650
653
654
660
661
664
665
666
667
670
671
672
673
675
678
679
690
697
701
702
704


705
707
708
720
738
744
746
747
762
767
771
776
782
786
793
822
851

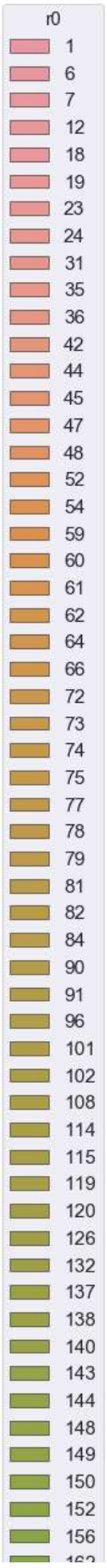
In [25]:

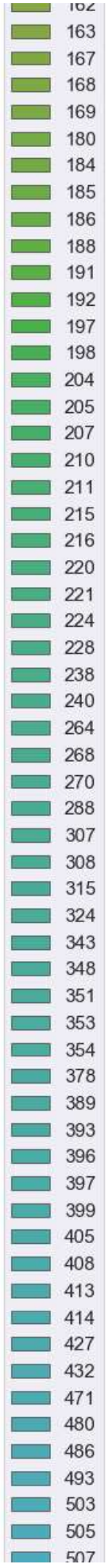
```
plt.figure(figsize=(12,12))
sns.boxplot(x='r0',y='time_intervals',data=dff, hue='target')
plt.show()
```

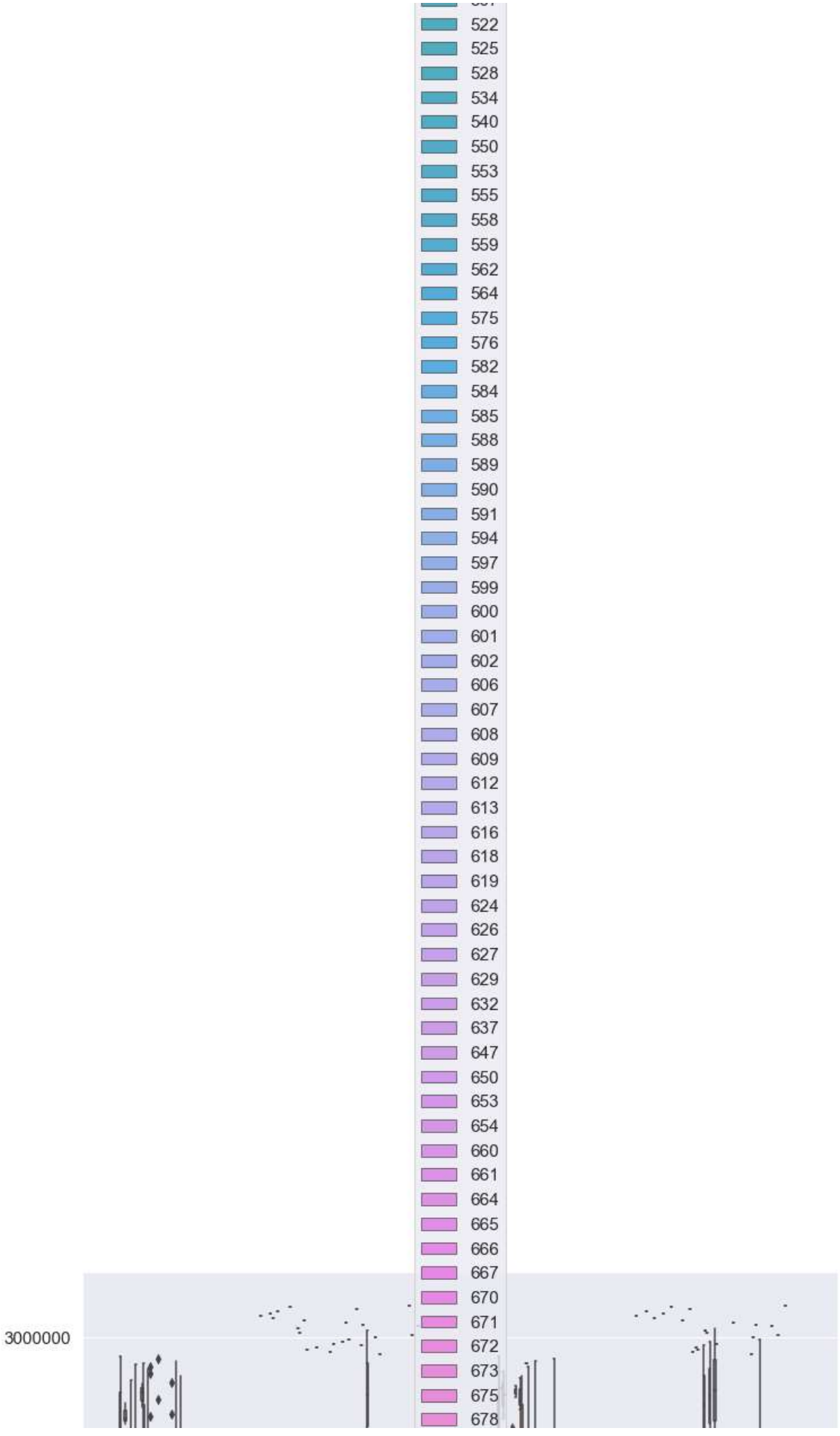

target
0
1
3000000
2500000
2000000
1500000
1000000
500000
0
time_intervals
r0

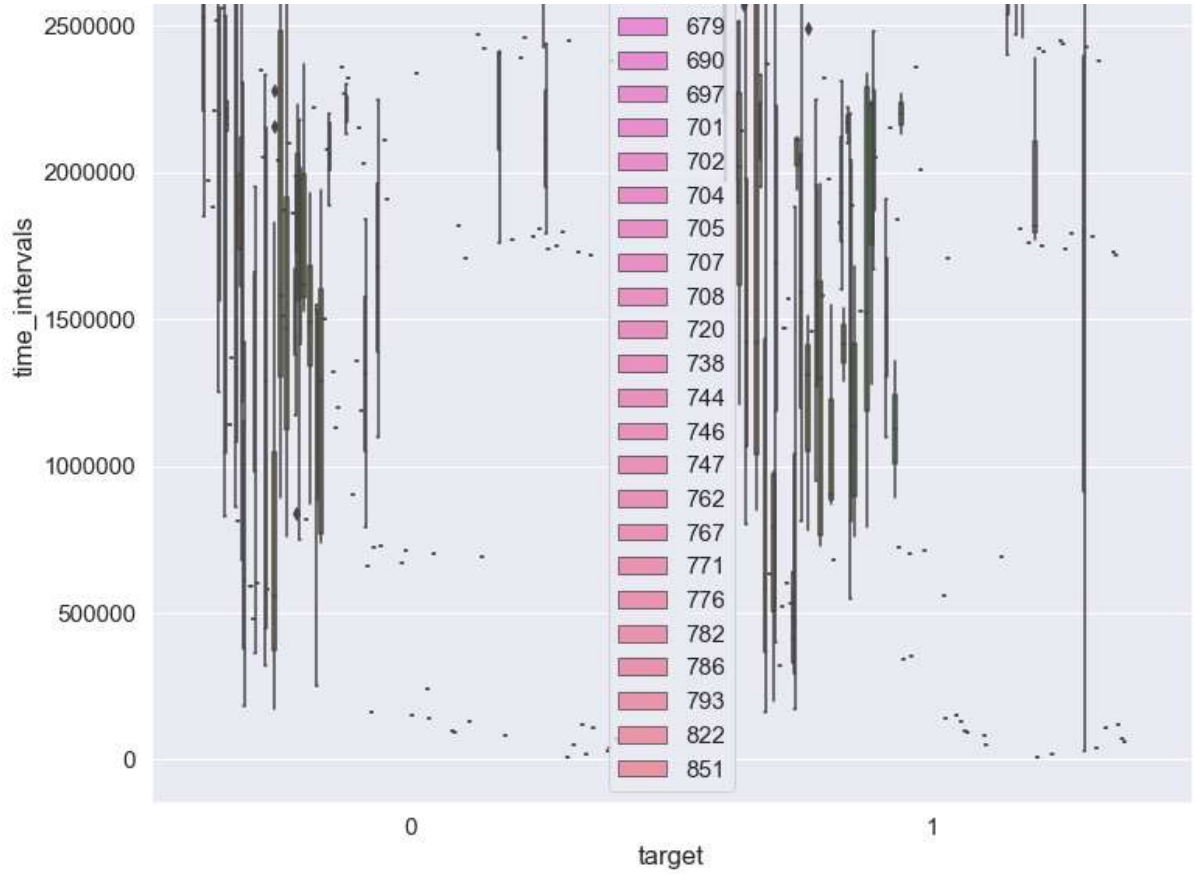
In [26]:

```
plt.figure(figsize=(12,12))
sns.boxplot(x='target',y='time_intervals',data=dff, hue='r0')
plt.show()
```


r0
1
6
7
12
18
19
23
24
31
35
36
42
44
45
47
48
52
54
59
60
61
62
64
66
72
73
74
75
77
78
79
81
82
84
90
91
96
101
102
108
114
115
119
120
126
132
137
138
140
143
144
148
149
150
152
156

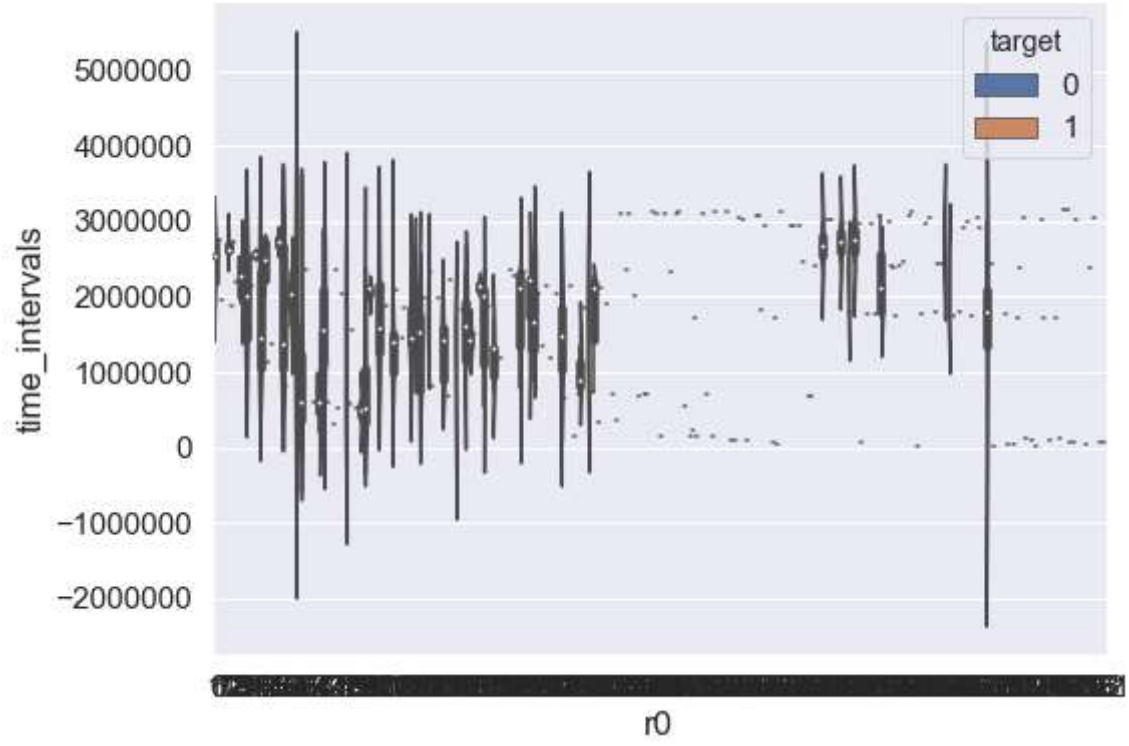

162
163
167
168
169
180
184
185
186
188
191
192
197
198
204
205
207
210
211
215
216
220
221
224
228
238
240
264
268
270
288
307
308
315
324
343
348
351
353
354
378
389
393
396
397
399
405
408
413
414
427
432
471
480
486
493
503
505
507


522
525
528
534
540
550
553
555
558
559
562
564
575
576
582
584
585
588
589
590
591
594
597
599
600
601
602
606
607
608
609
612
613
616
618
619
624
626
627
629
632
637
647
650
653
654
660
661
664
665
666
667
670
671
672
673
675
678
3000000



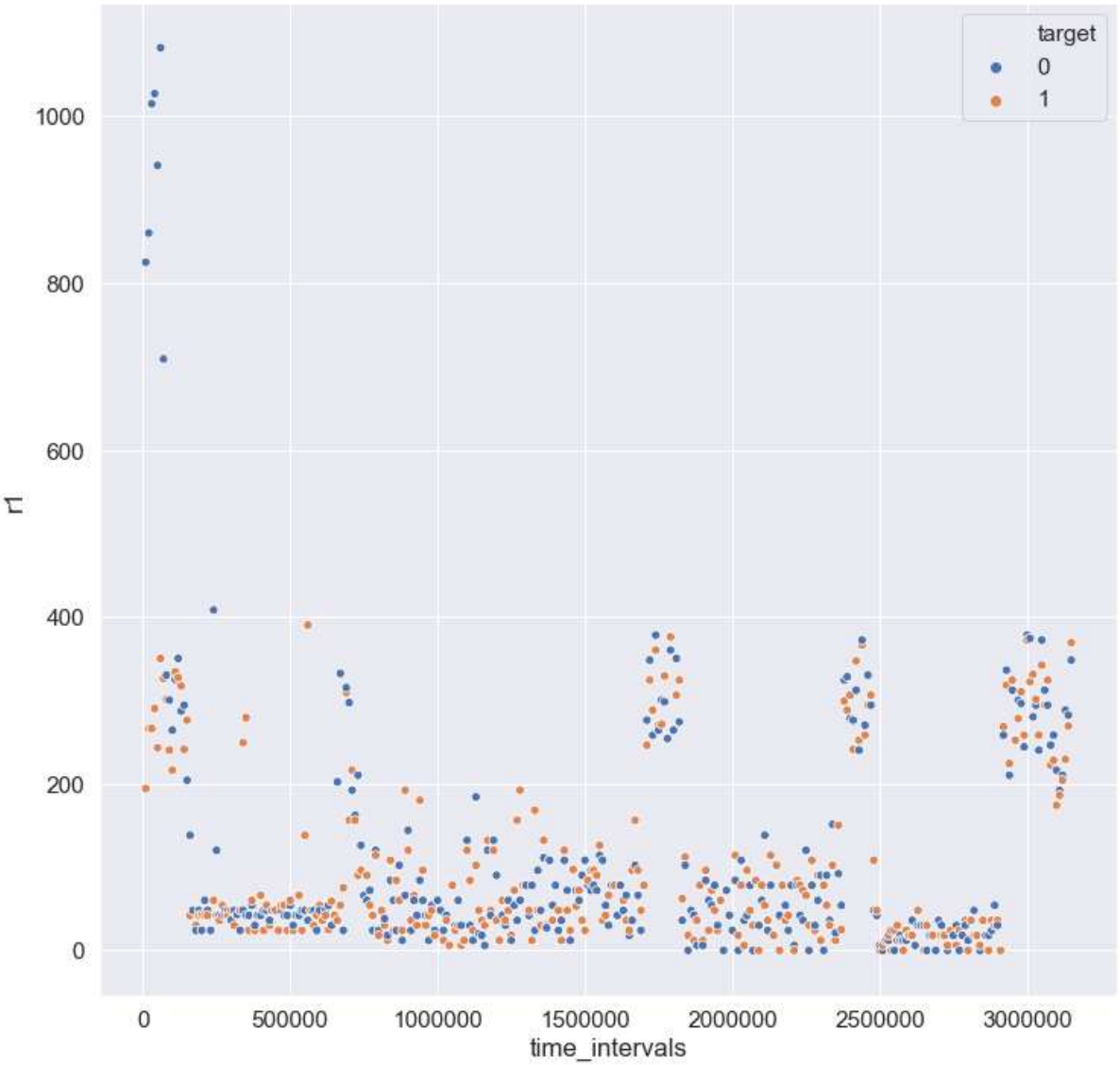
In [27]:

```
plt.figure(figsize=(8,6))
sns.violinplot(x='r0',y='time_intervals',data=dff, hue='target', split=True)
plt.show()
```



In [28]:

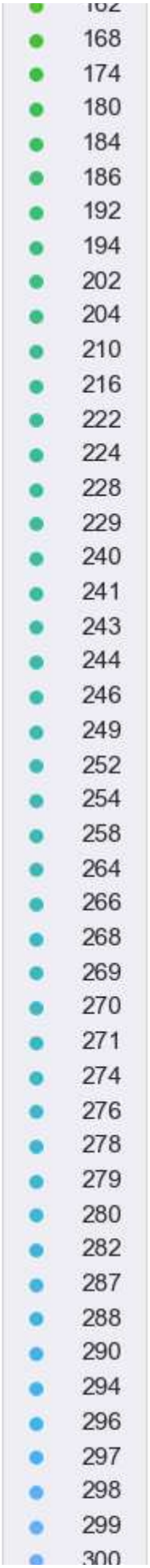
```
plt.figure(figsize=(12,12))
sns.scatterplot(x='time_intervals',y='r1',data=dff, hue='target')
plt.show()
```

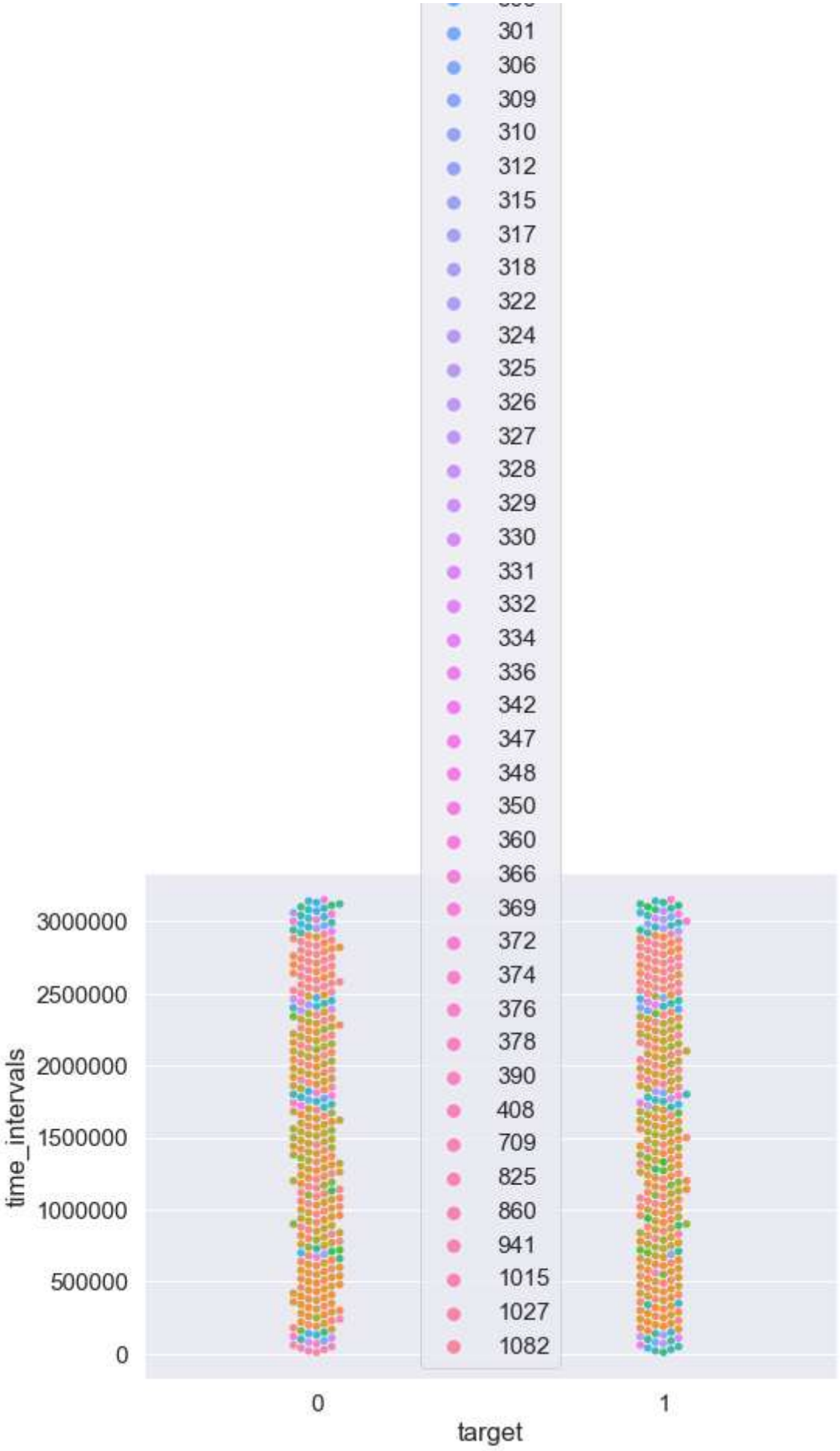


In [29]:

```
plt.figure(figsize=(8,6))
sns.swarmplot(x='target',y='time_intervals',data=dff, hue='r1')
plt.show()
```

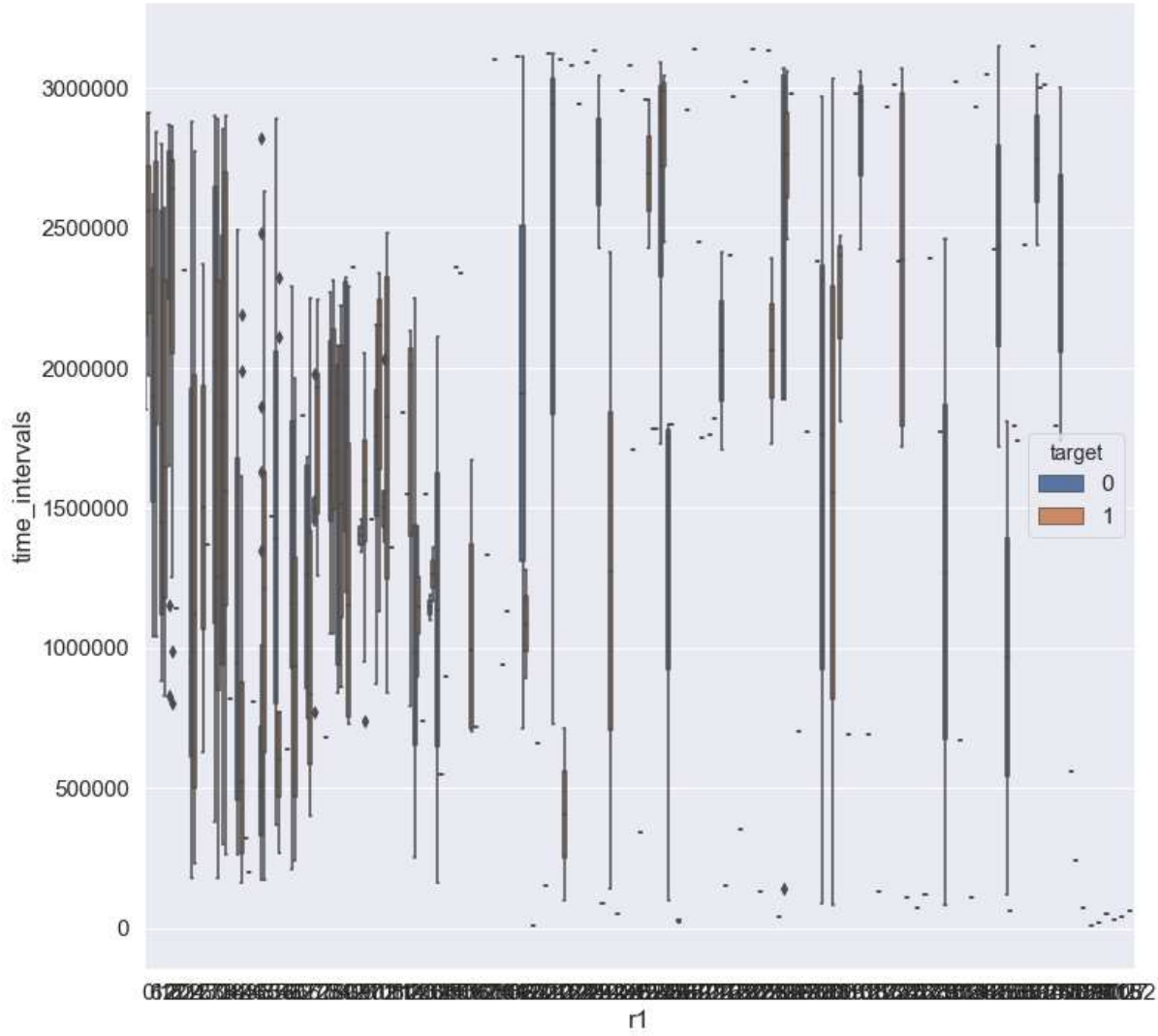

r1
0
6
12
18
20
21
24
25
27
30
36
38
42
43
46
48
53
54
59
60
62
66
72
75
78
84
90
92
96
97
102
108
111
112
114
120
126
132
138
144
150
151
156


168
174
180
184
186
192
194
202
204
210
216
222
224
228
229
240
241
243
244
246
249
252
254
258
264
266
268
269
270
271
274
276
278
279
280
282
287
288
290
294
296
297
298
299
300


301
306
309
310
312
315
317
318
322
324
325
326
327
328
329
330
331
332
334
336
342
347
348
350
360
366
369
372
374
376
378
390
408
709
825
860
941
1015
1027
1082
3000000
2500000
2000000
1500000
1000000
500000
0
time_intervals
0
1
target

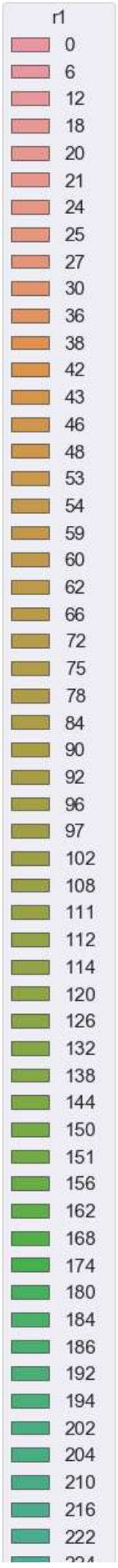
In [30]:

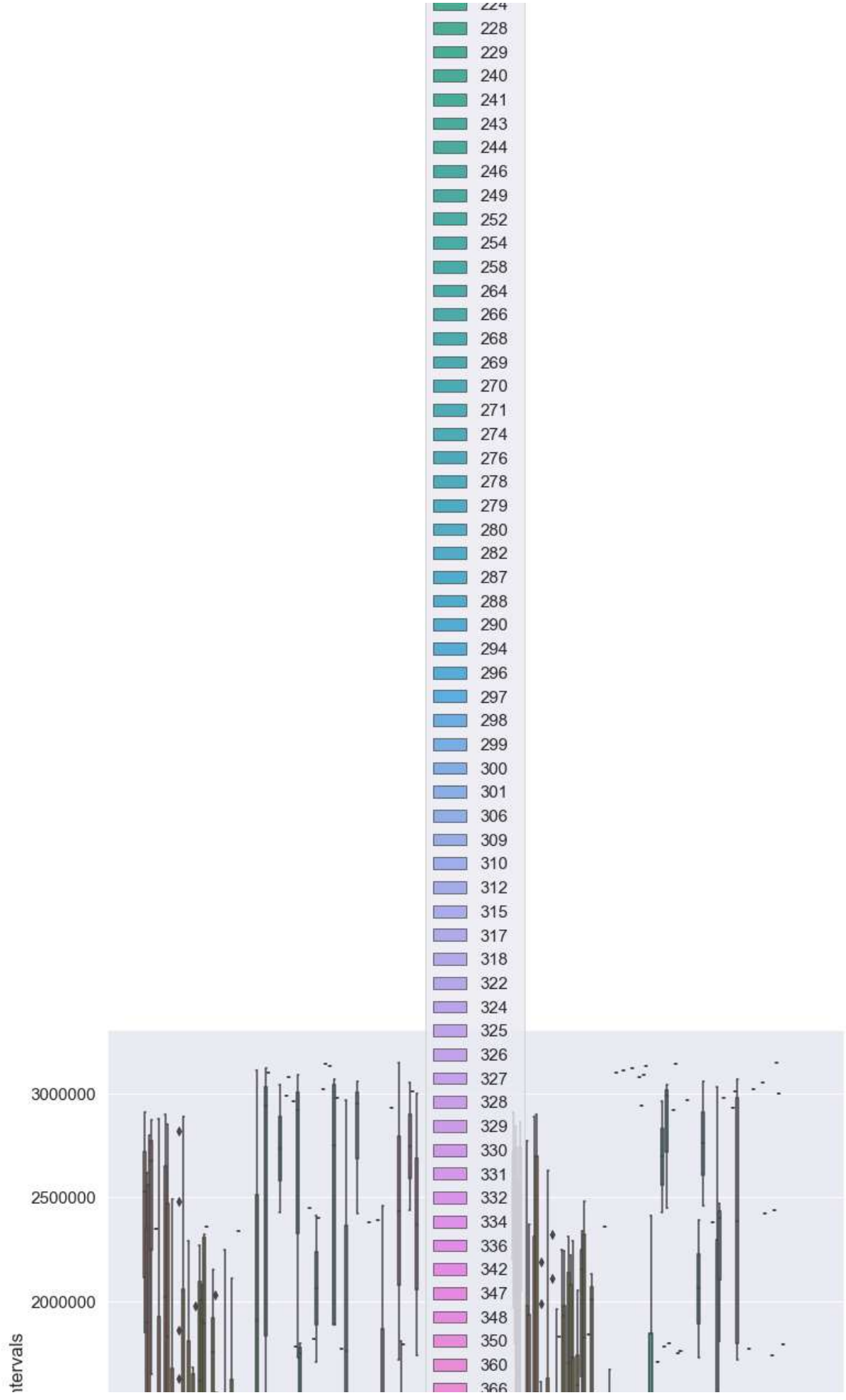
```
plt.figure(figsize=(12,12))
sns.boxplot(x='r1',y='time_intervals',data=dff, hue='target')
plt.show()
```

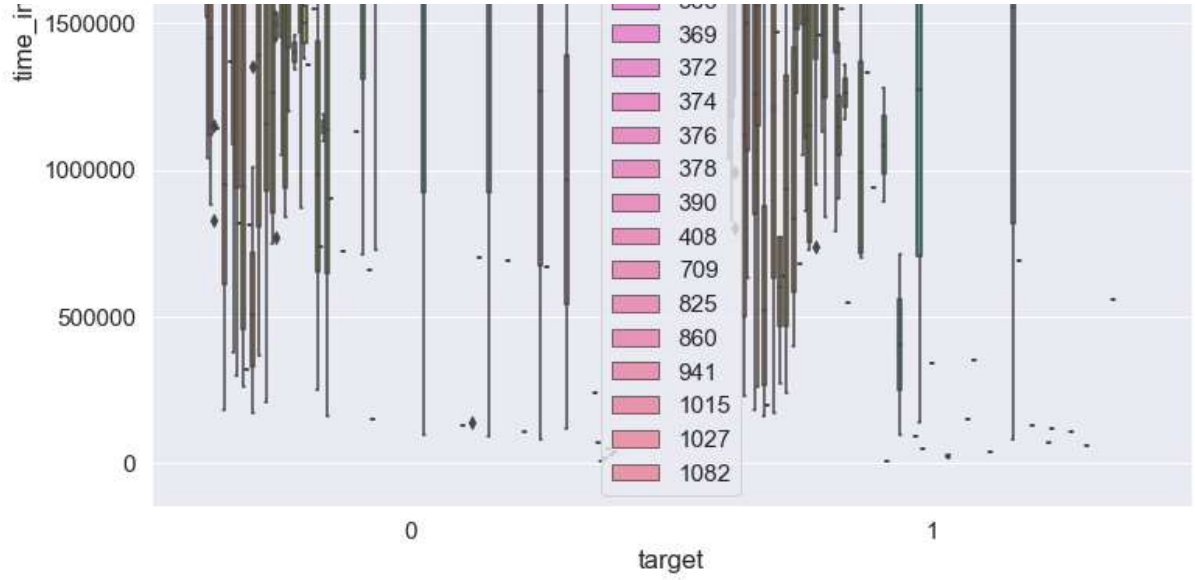


In [31]:

```
plt.figure(figsize=(12,12))
sns.boxplot(x='target',y='time_intervals',data=dff, hue='r1')
plt.show()
```

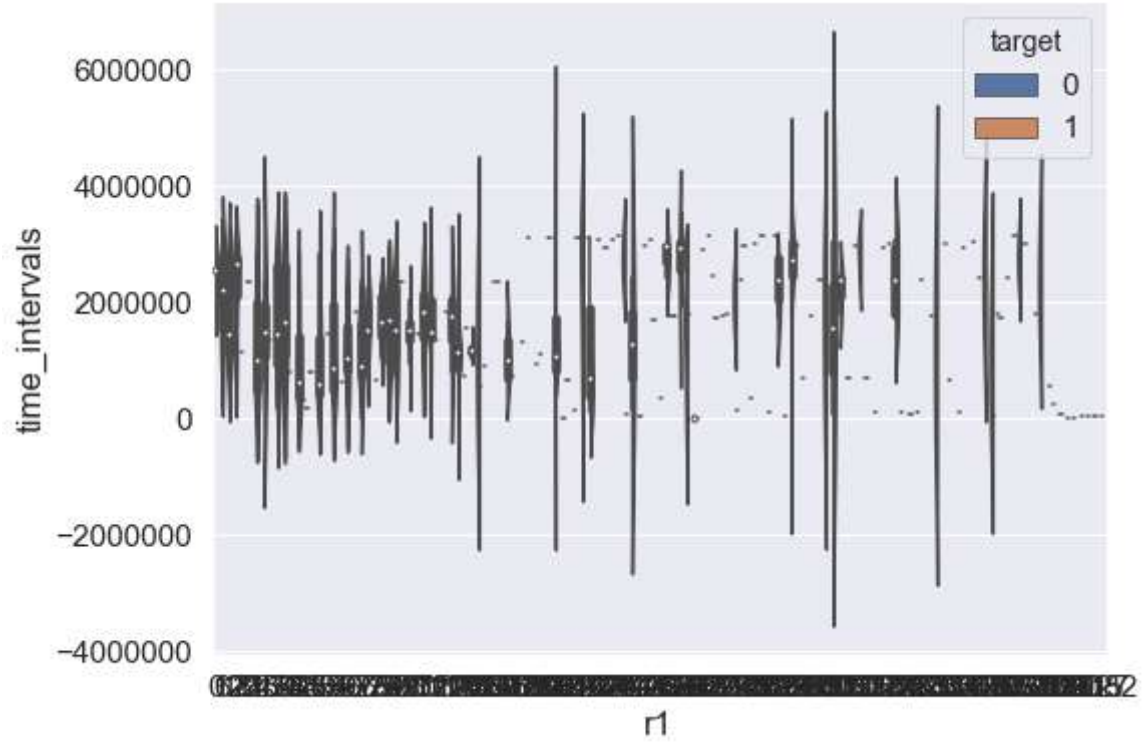

r1
0
6
12
18
20
21
24
25
27
30
36
38
42
43
46
48
53
54
59
60
62
66
72
75
78
84
90
92
96
97
102
108
111
112
114
120
126
132
138
144
150
151
156
162
168
174
180
184
186
192
194
202
204
210
216
222


224
228
229
240
241
243
244
246
249
252
254
258
264
266
268
269
270
271
274
276
278
279
280
282
287
288
290
294
296
297
298
299
300
301
306
309
310
312
315
317
318
322
324
325
326
327
328
329
330
331
332
334
336
342
347
348
350
360
366
3000000
2500000
2000000
tervals



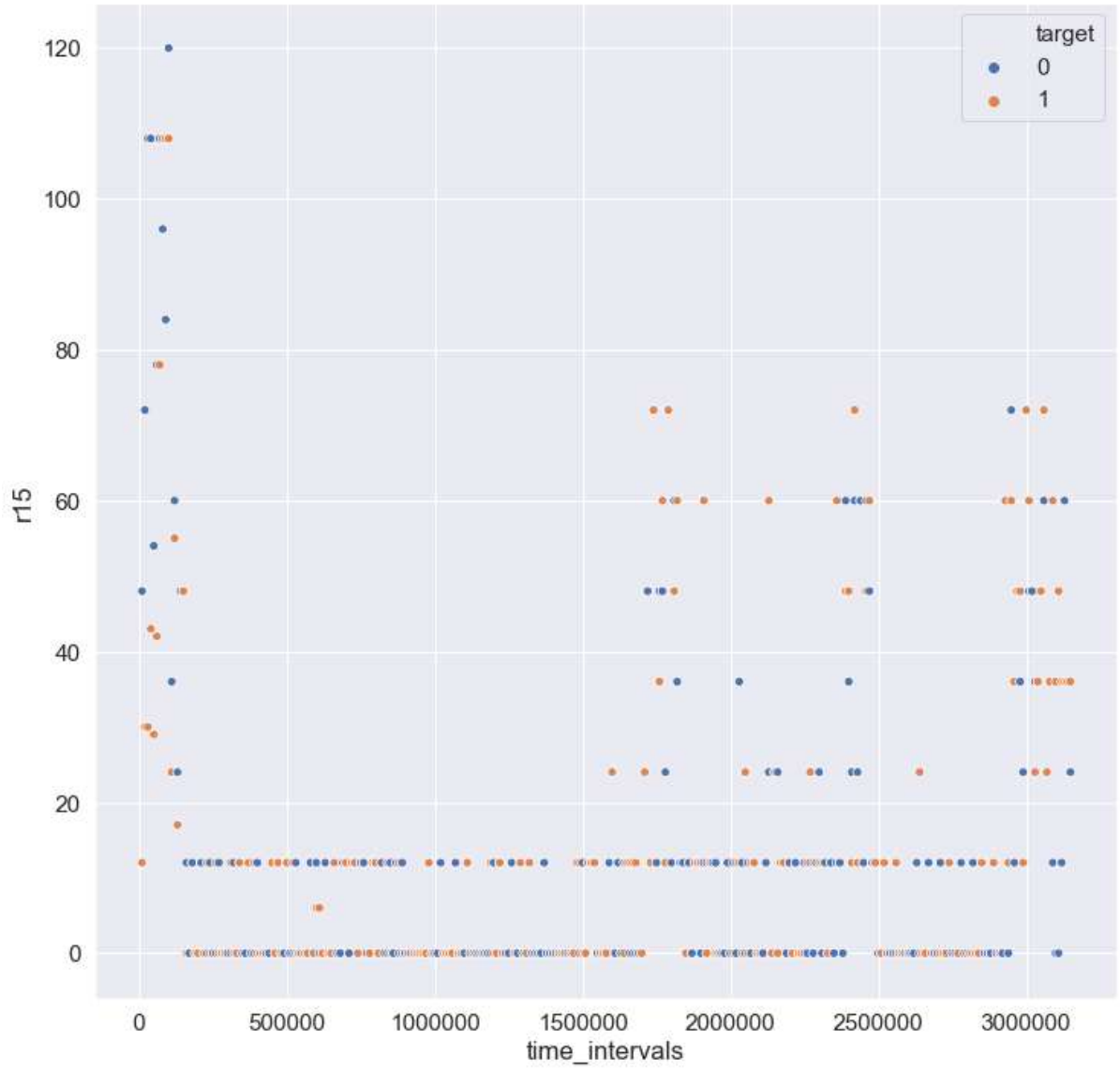
In [32]:

```
plt.figure(figsize=(8,6))
sns.violinplot(x='r1',y='time_intervals',data=dff, hue='target', split=True)
plt.show()
```



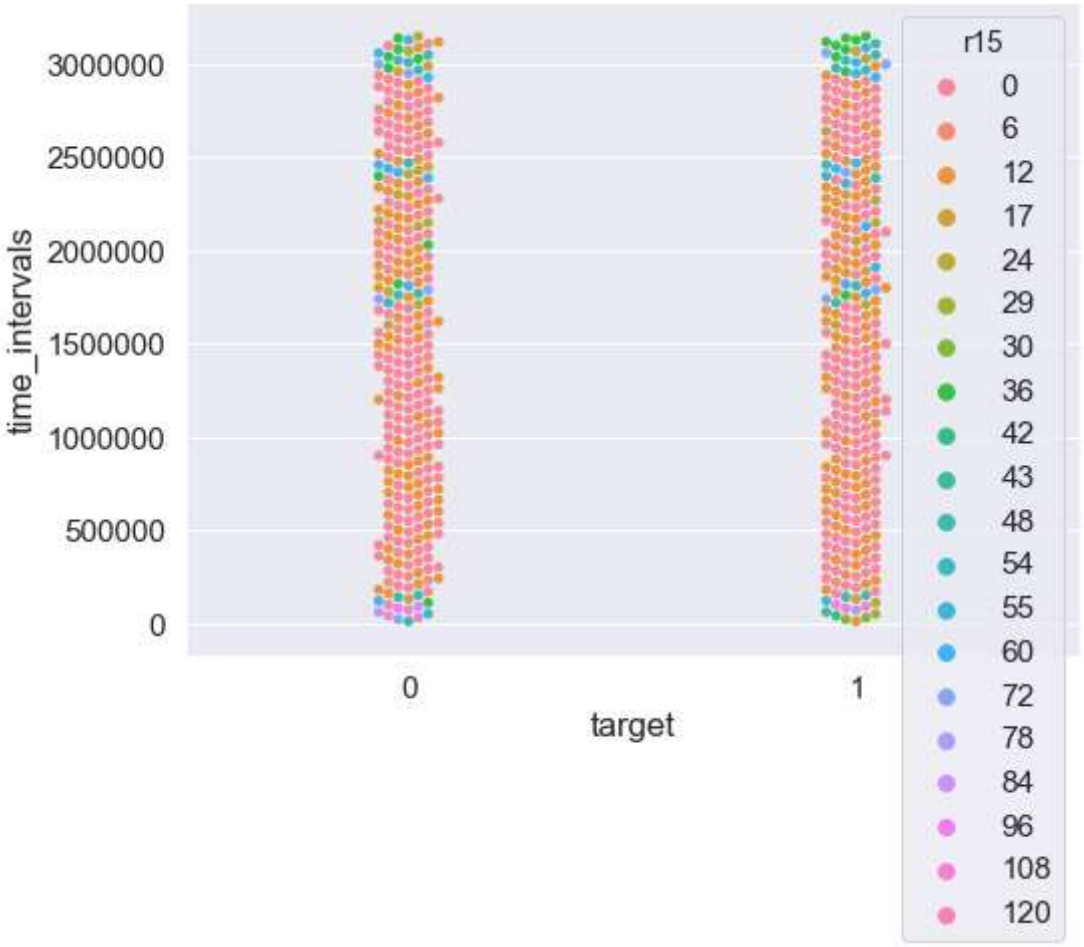
In [33]:

```
plt.figure(figsize=(12,12))
sns.scatterplot(x='time_intervals',y='r15',data=dff, hue='target')
plt.show()
```



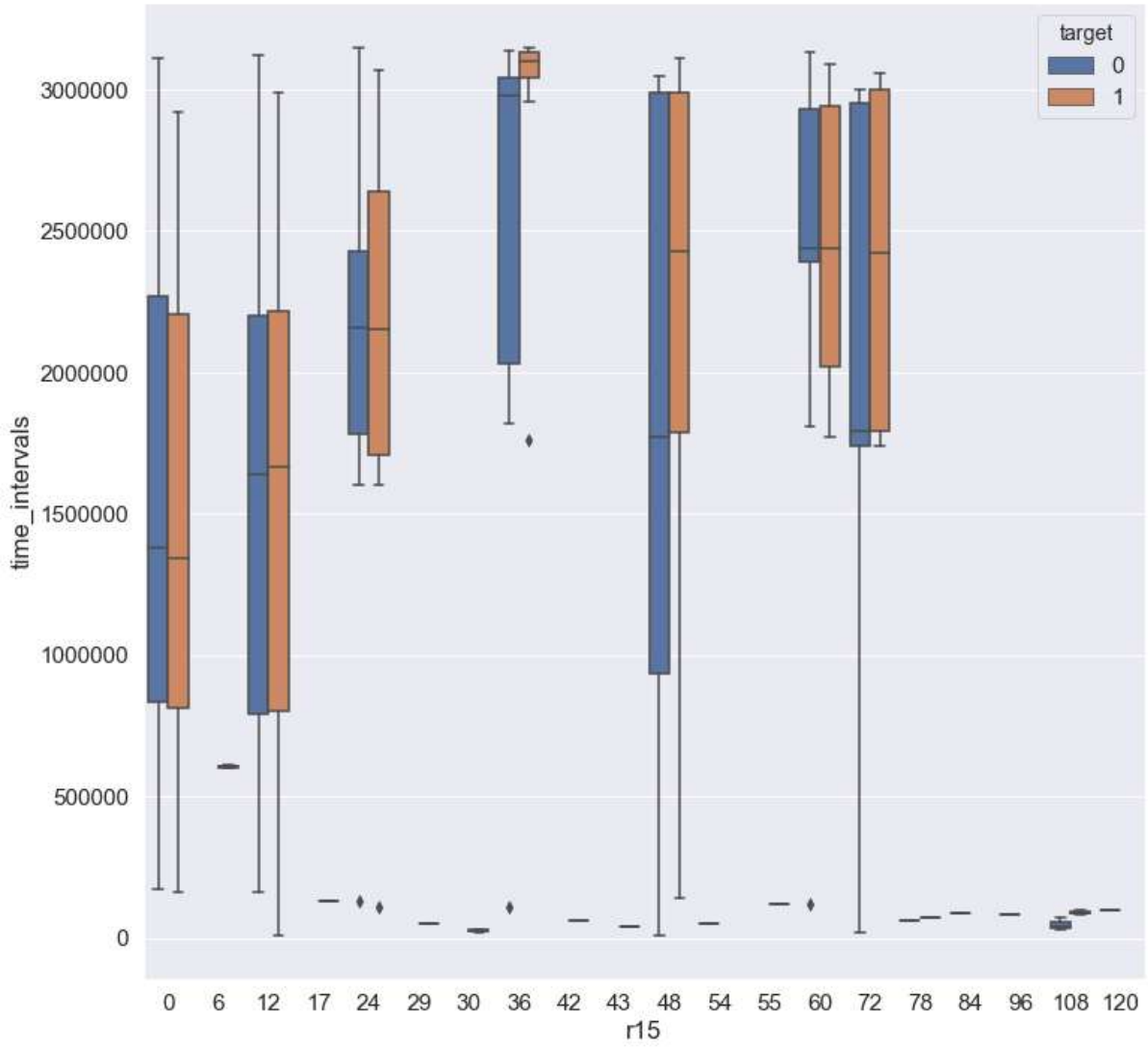
In [34]:

```
plt.figure(figsize=(8,6))
sns.swarmplot(x='target',y='time_intervals',data=dff, hue='r15')
plt.show()
```



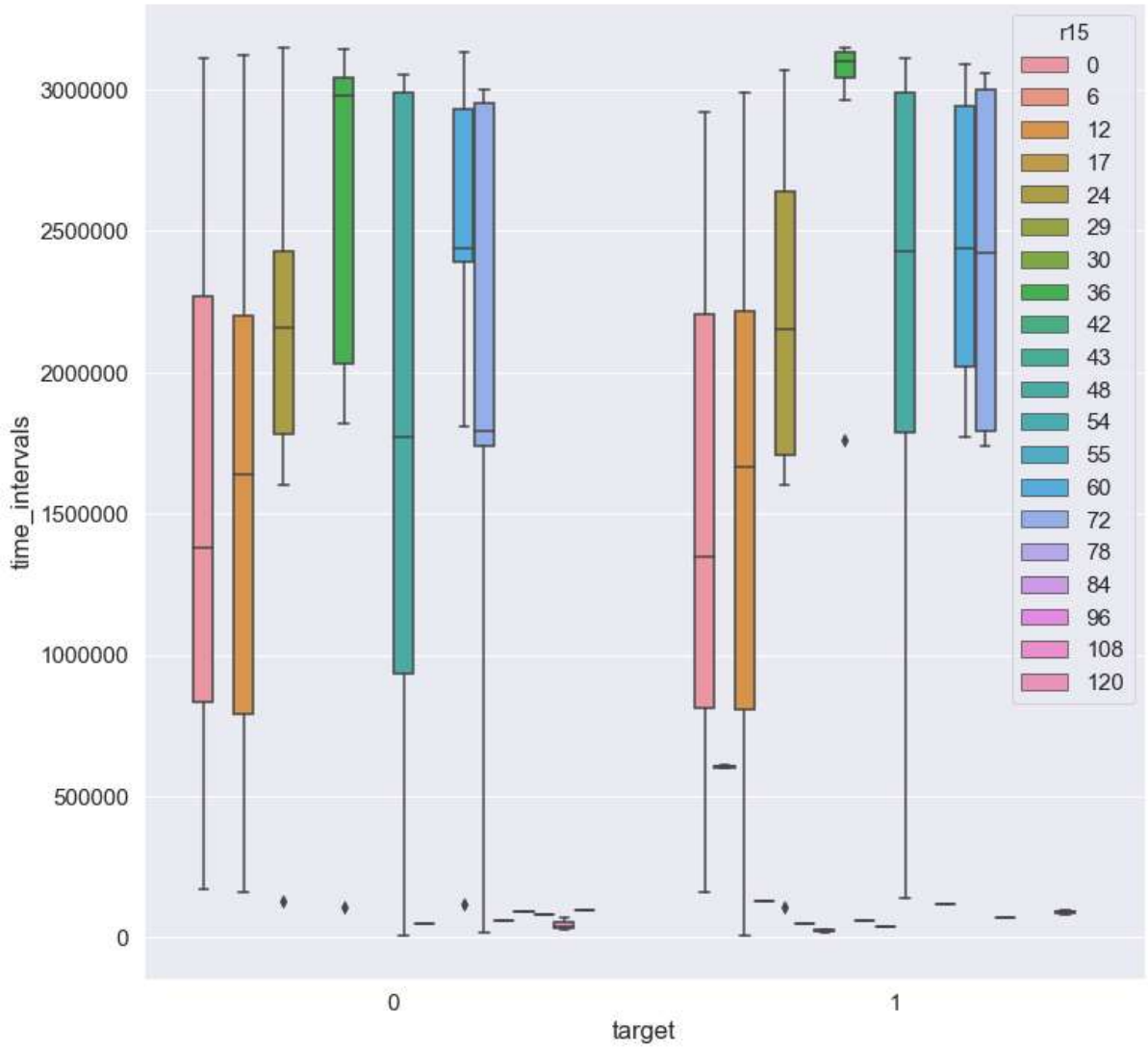
In [35]:

```
plt.figure(figsize=(12,12))
sns.boxplot(x='r15',y='time_intervals',data=dff, hue='target')
plt.show()
```



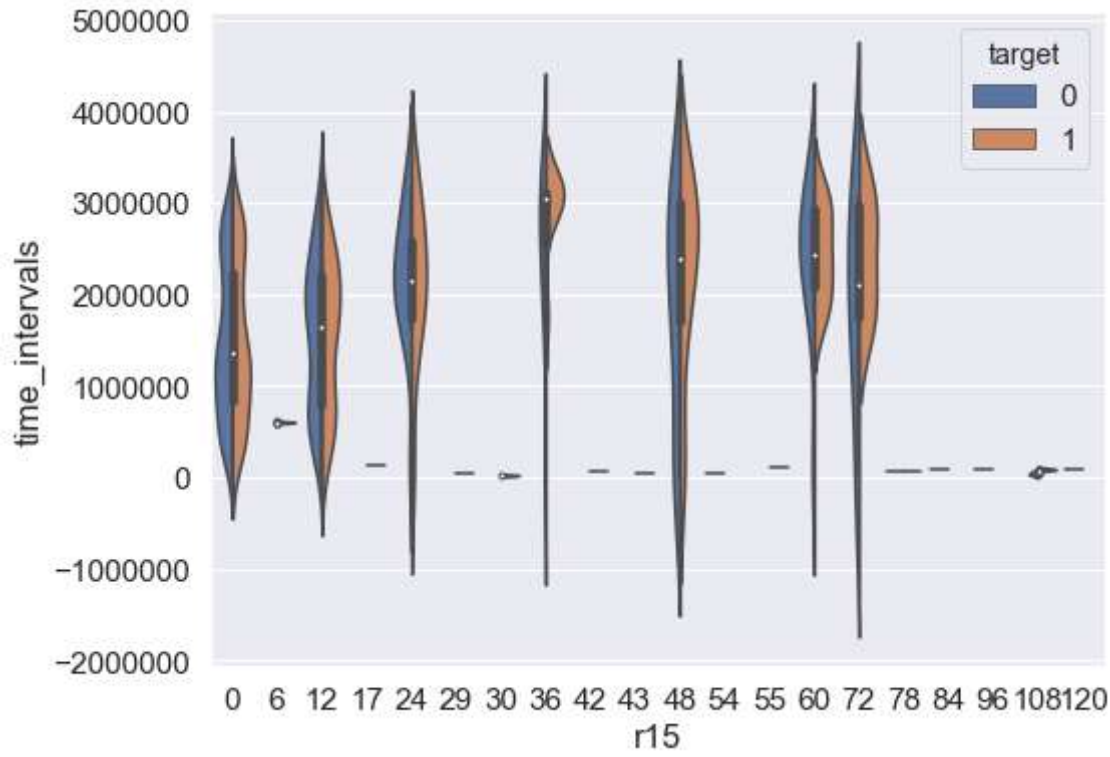
In [36]:

```
plt.figure(figsize=(12,12))
sns.boxplot(x='target',y='time_intervals',data=dff, hue='r15')
plt.show()
```



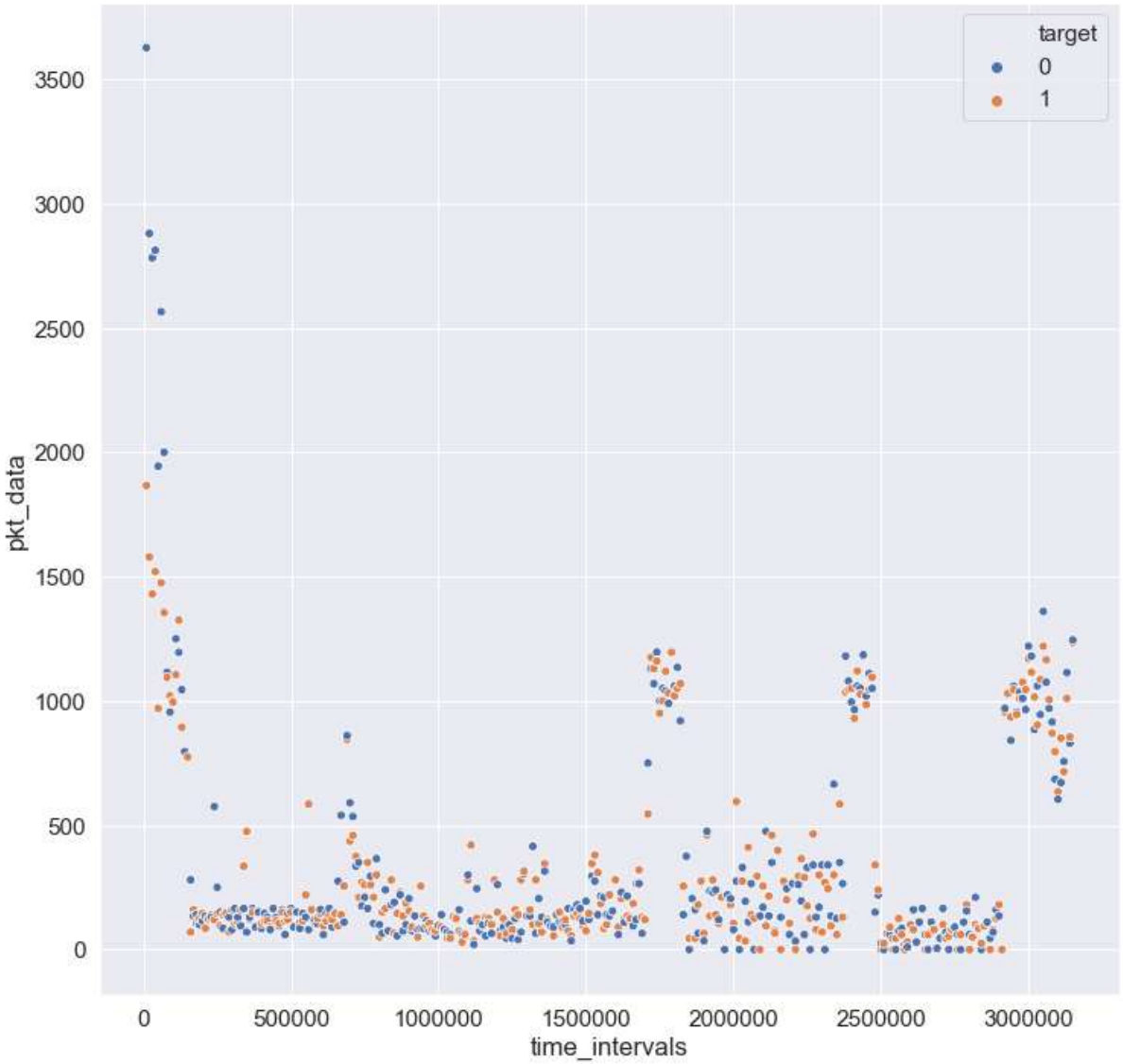
In [37]:

```
plt.figure(figsize=(8,6))
sns.violinplot(x='r15',y='time_intervals',data=dff, hue='target', split=True)
plt.show()
```



In [38]:

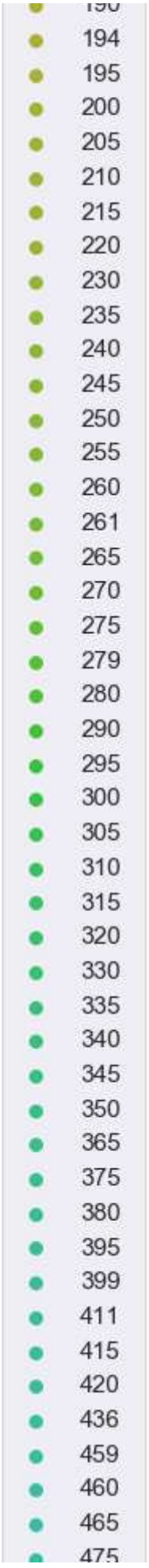
```
plt.figure(figsize=(12,12))
sns.scatterplot(x='time_intervals',y='pkt_data',data=dff, hue='target')
plt.show()
```

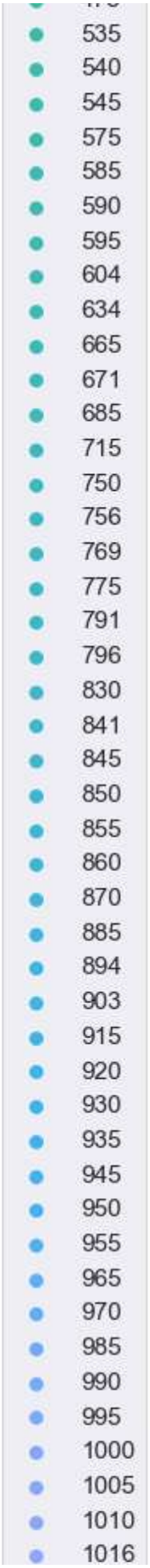


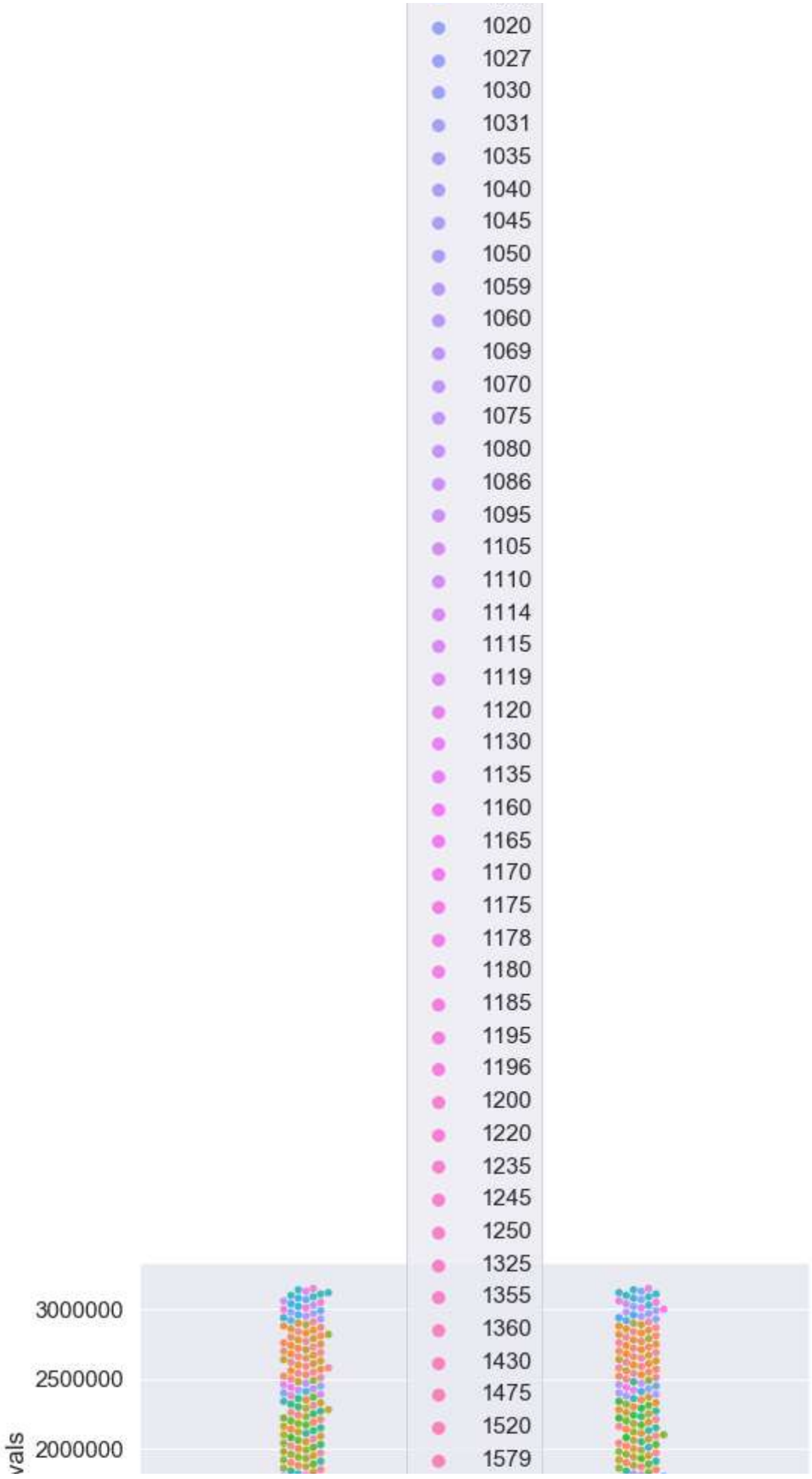
In [39]:

```
plt.figure(figsize=(8,6))
sns.swarmplot(x='target',y='time_intervals',data=dff, hue='pkt_data')
plt.show()
```


pkt_data
0
5
10
20
25
30
34
35
40
45
50
55
60
65
66
70
75
80
85
90
95
100
104
105
110
111
115
120
121
125
130
135
140
145
150
155
160
161
165
170
175
180
185

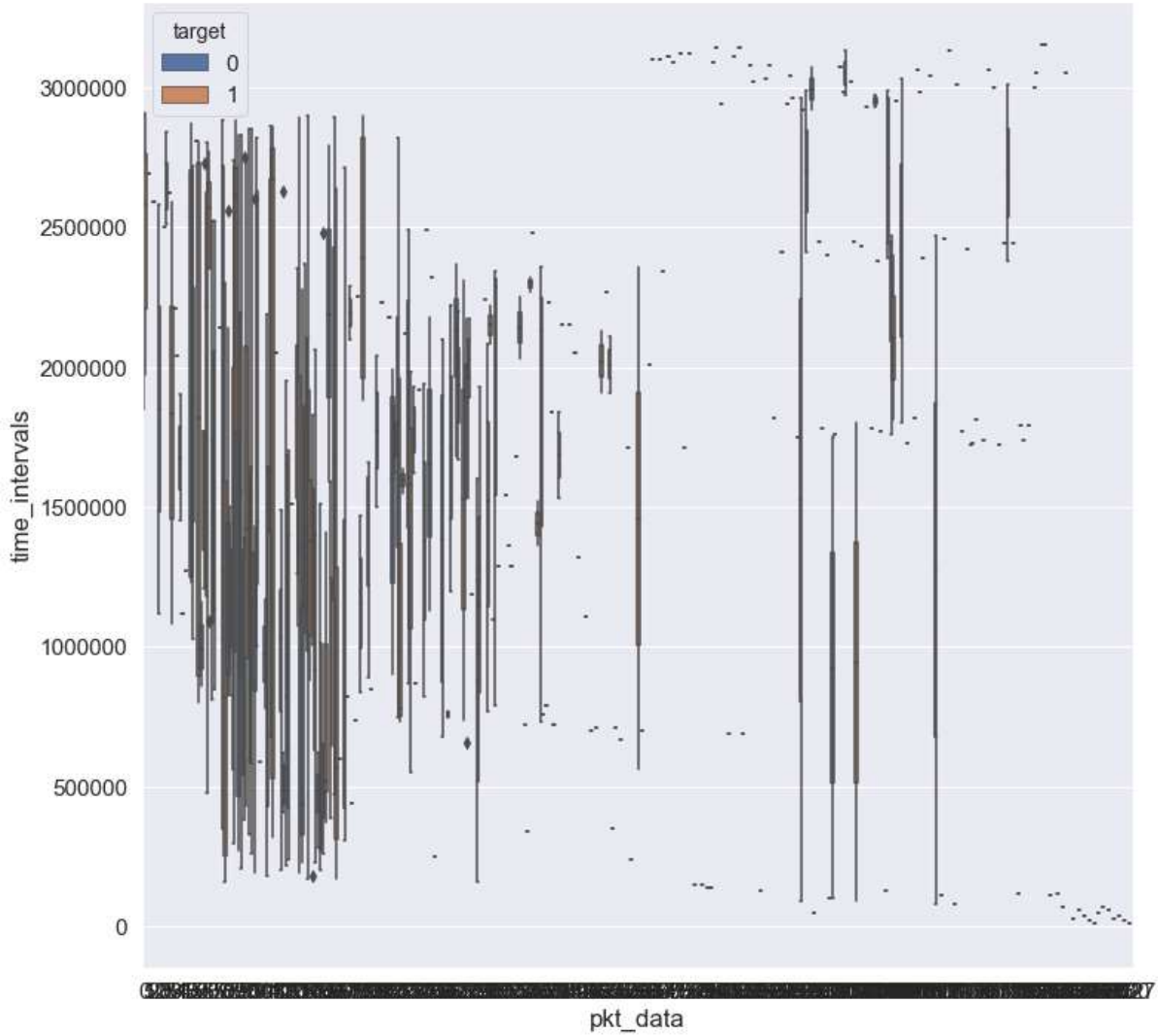

194
195
200
205
210
215
220
230
235
240
245
250
255
260
261
265
270
275
279
280
290
295
300
305
310
315
320
330
335
340
345
350
365
375
380
395
399
411
415
420
436
459
460
465
475


535
540
545
575
585
590
595
604
634
665
671
685
715
750
756
769
775
791
796
830
841
845
850
855
860
870
885
894
903
915
920
930
935
945
950
955
965
970
985
990
995
1000
1005
1010
1016


1020
1027
1030
1031
1035
1040
1045
1050
1059
1060
1069
1070
1075
1080
1086
1095
1105
1110
1114
1115
1119
1120
1130
1135
1160
1165
1170
1175
1178
1180
1185
1195
1196
1200
1220
1235
1245
1250
1325
1355
1360
1430
1475
1520
1579
3000000
2500000
2000000
vals

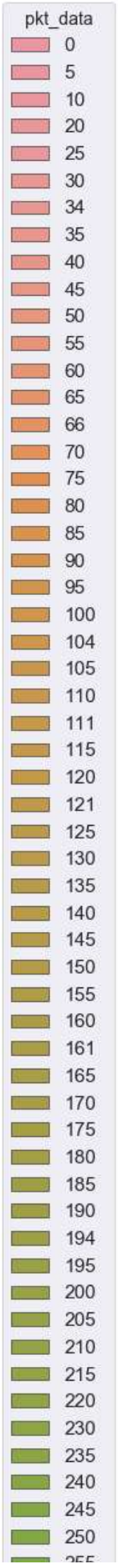
In [40]:

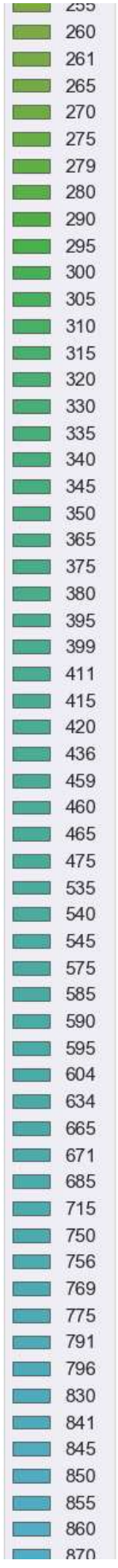
```
plt.figure(figsize=(12,12))
sns.boxplot(x='pkt_data',y='time_intervals',data=dff, hue='target')
plt.show()
```

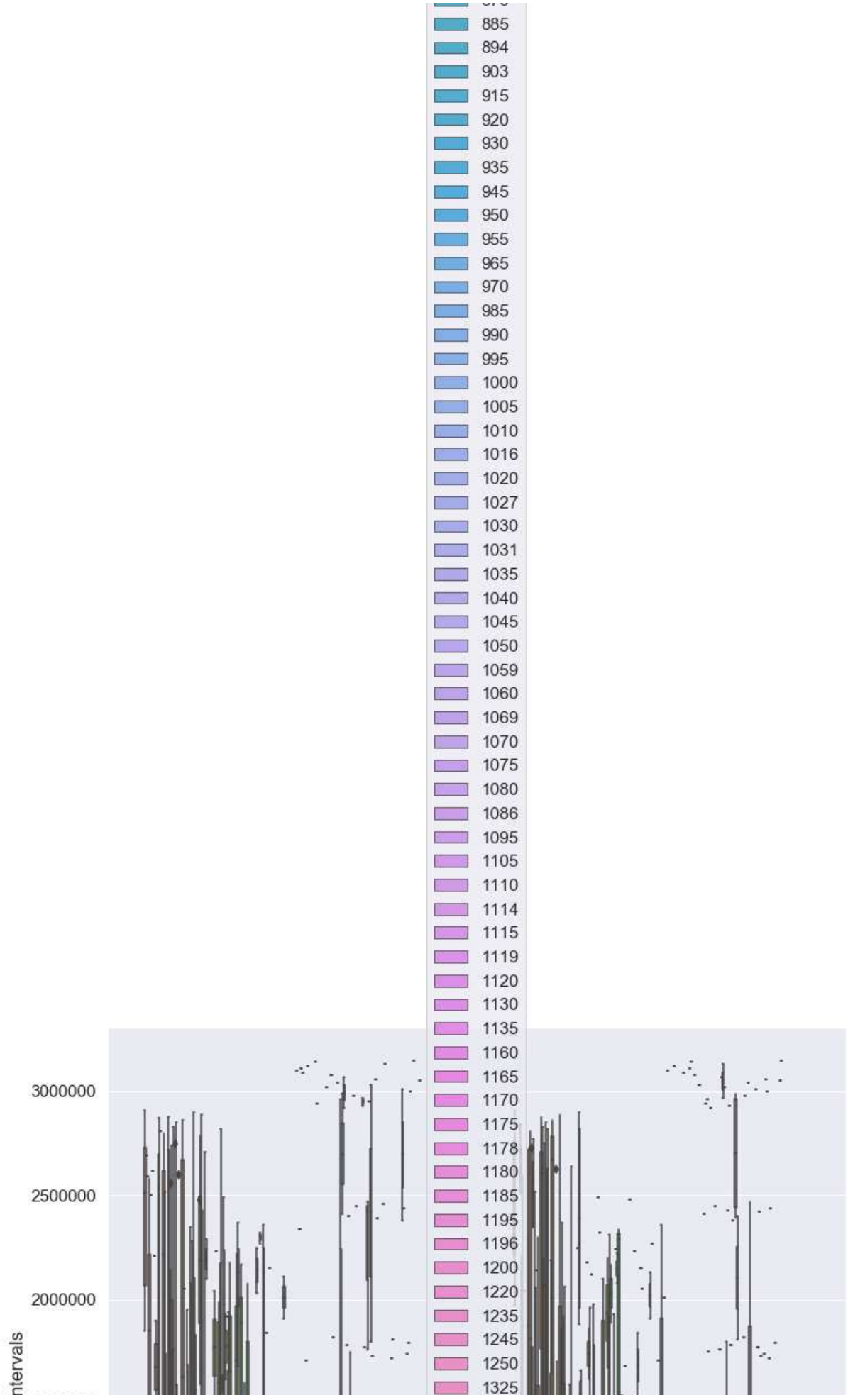


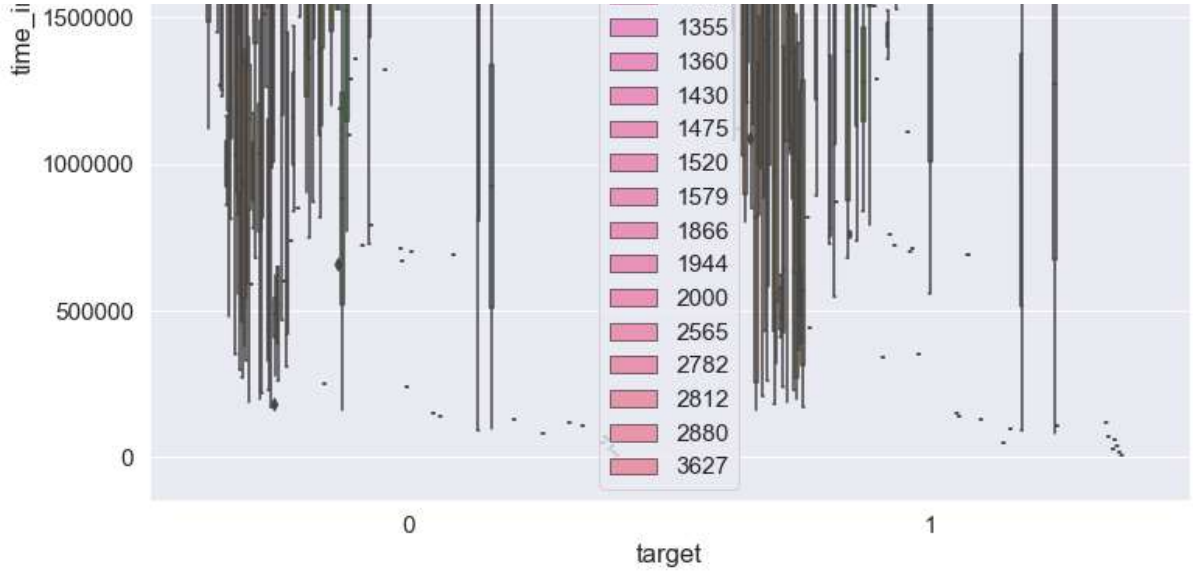
In [41]:

```
plt.figure(figsize=(12,12))
sns.boxplot(x='target',y='time_intervals',data=dff, hue='pkt_data')
plt.show()
```


pkt_data
0
5
10
20
25
30
34
35
40
45
50
55
60
65
66
70
75
80
85
90
95
100
104
105
110
111
115
120
121
125
130
135
140
145
150
155
160
161
165
170
175
180
185
190
194
195
200
205
210
215
220
230
235
240
245
250

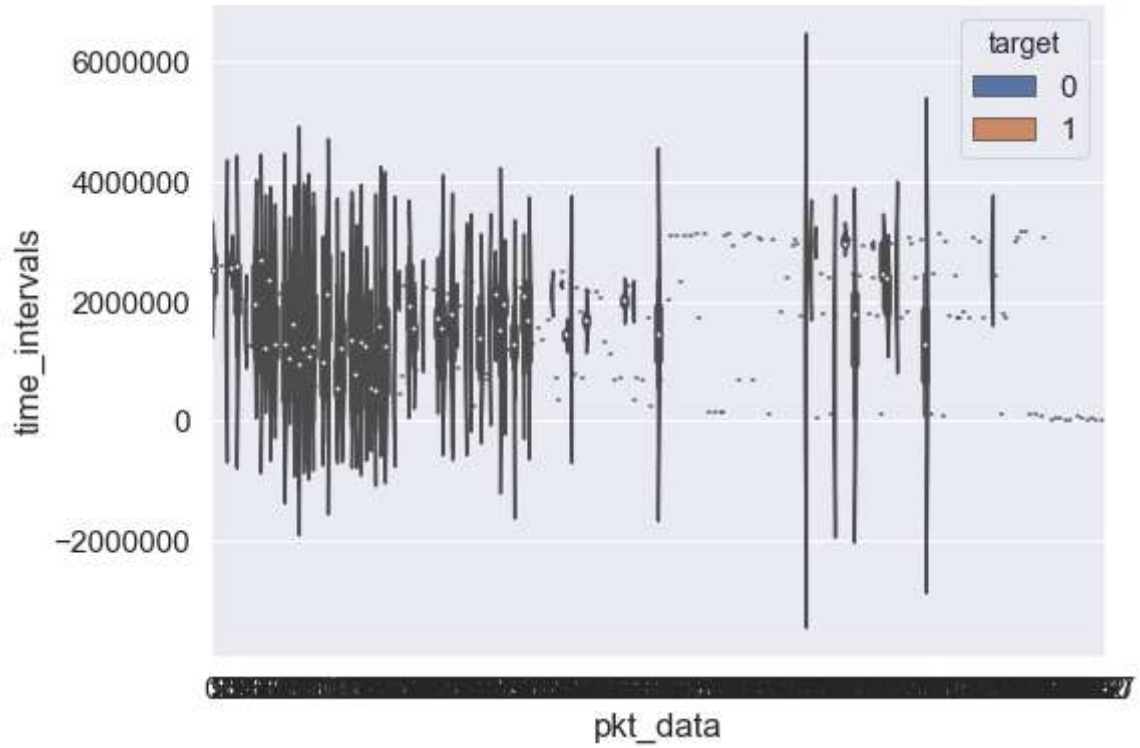

255
260
261
265
270
275
279
280
290
295
300
305
310
315
320
330
335
340
345
350
365
375
380
395
399
411
415
420
436
459
460
465
475
535
540
545
575
585
590
595
604
634
665
671
685
715
750
756
769
775
791
796
830
841
845
850
855
860
870


885
894
903
915
920
930
935
945
950
955
965
970
985
990
995
1000
1005
1010
1016
1020
1027
1030
1031
1035
1040
1045
1050
1059
1060
1069
1070
1075
1080
1086
1095
1105
1110
1114
1115
1119
1120
1130
1135
1160
1165
1170
1175
1178
1180
1185
1195
1196
1200
1220
1235
1245
1250
1325
3000000
2500000
2000000
ntervals



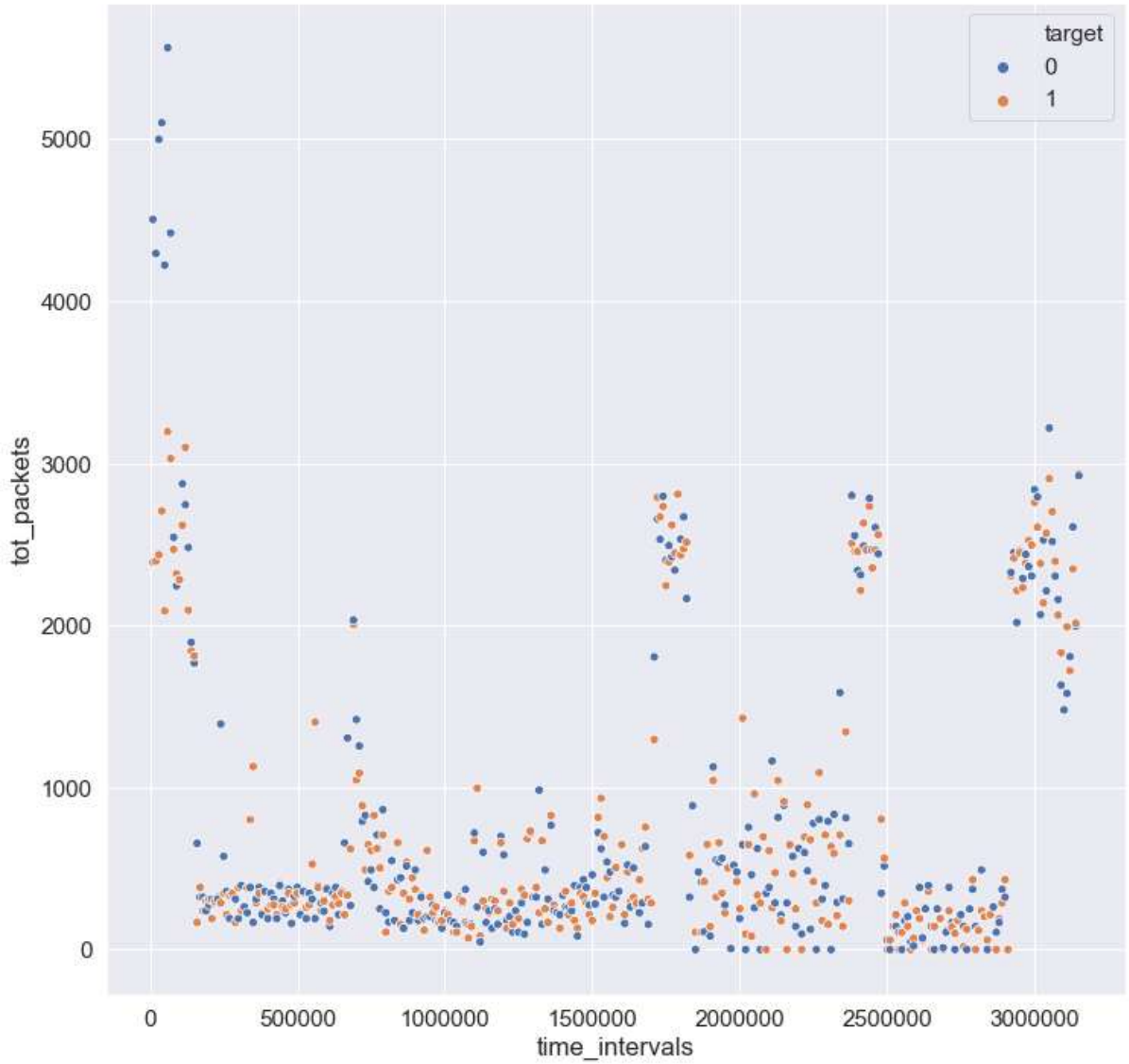
In [42]:

```
plt.figure(figsize=(8,6))
sns.violinplot(x='pkt_data',y='time_intervals',data=dff, hue='target', split=True)
plt.show()
```



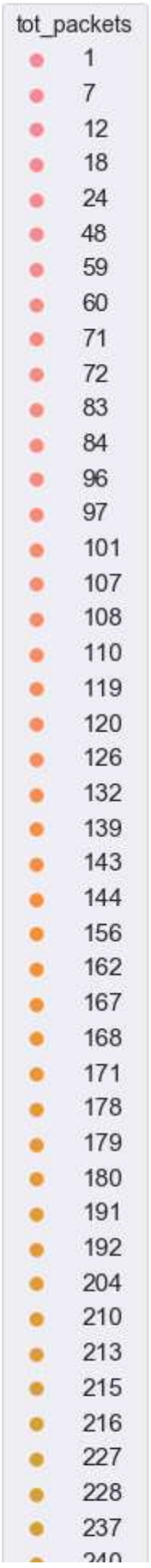
In [43]:

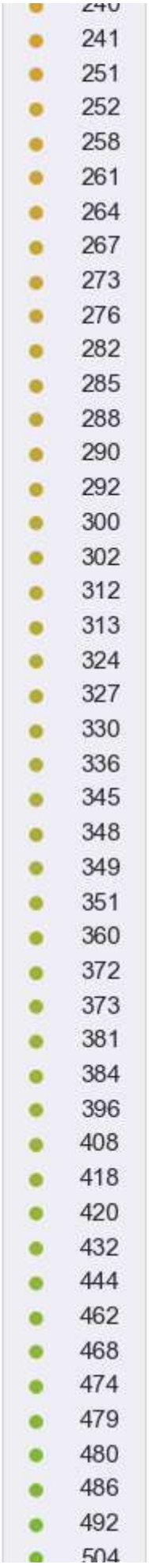
```
plt.figure(figsize=(12,12))
sns.scatterplot(x='time_intervals',y='tot_packets',data=dff, hue='target')
plt.show()
```

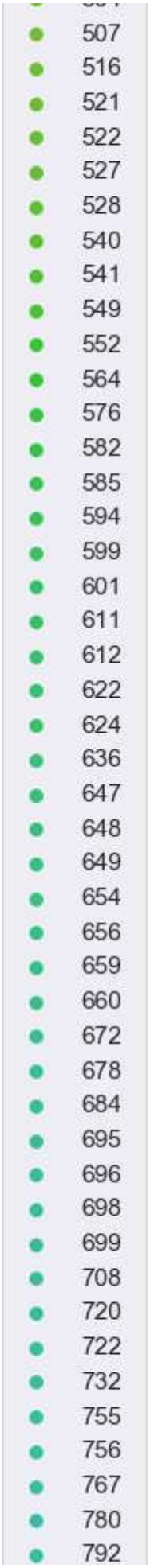


In [44]:

```
plt.figure(figsize=(8,6))
sns.swarmplot(x='target',y='time_intervals',data=dff, hue='tot_packets')
plt.show()
```

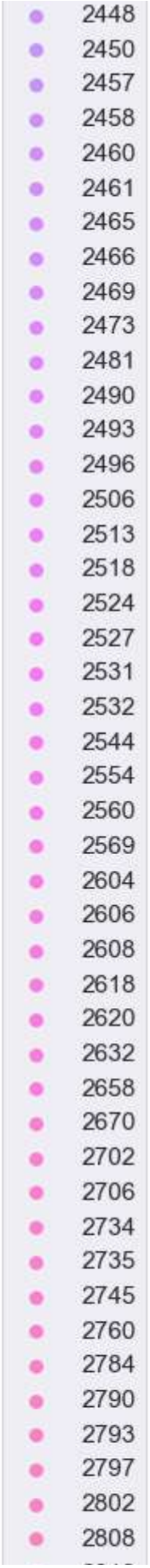

tot_packets
1
7
12
18
24
48
59
60
71
72
83
84
96
97
101
107
108
110
119
120
126
132
139
143
144
156
162
167
168
171
178
179
180
191
192
204
210
213
215
216
227
228
237

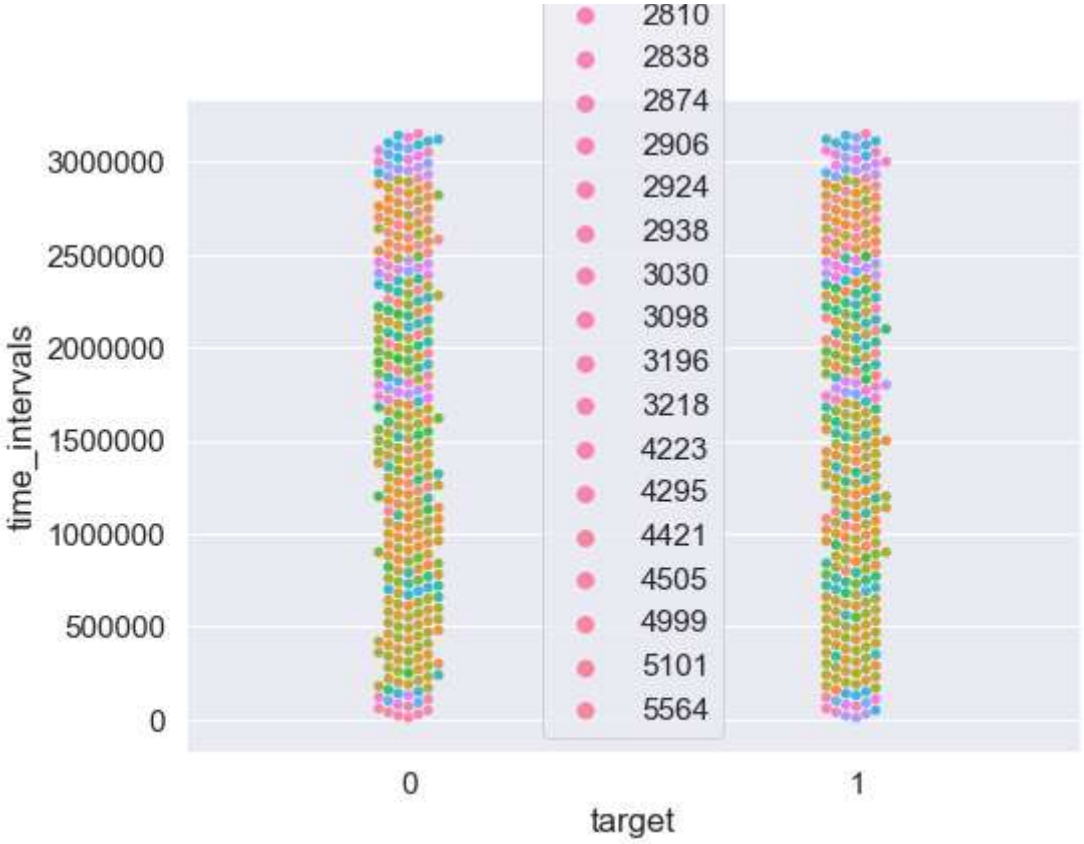

240
241
251
252
258
261
264
267
273
276
282
285
288
290
292
300
302
312
313
324
327
330
336
345
348
349
351
360
372
373
381
384
396
408
418
420
432
444
462
468
474
479
480
486
492
504


507
516
521
522
527
528
540
541
549
552
564
576
582
585
594
599
601
611
612
622
624
636
647
648
649
654
656
659
660
672
678
684
695
696
698
699
708
720
722
732
755
756
767
780
792

- 802
- 803
- 804
- 813
- 816
- 828
- 835
- 864
- 888
- 891
- 894
- 914
- 934
- 961
- 984
- 996
- 1044
- 1047
- 1089
- 1092
- 1128
- 1130
- 1164
- 1257
- 1296
- 1306
- 1344
- 1392
- 1404
- 1420
- 1428
- 1479
- 1480
- 1581
- 1586
- 1632
- 1721
- 1768
- 1806
- 1808
- 1812
- 1832
- 1842
- 1896
- 1992

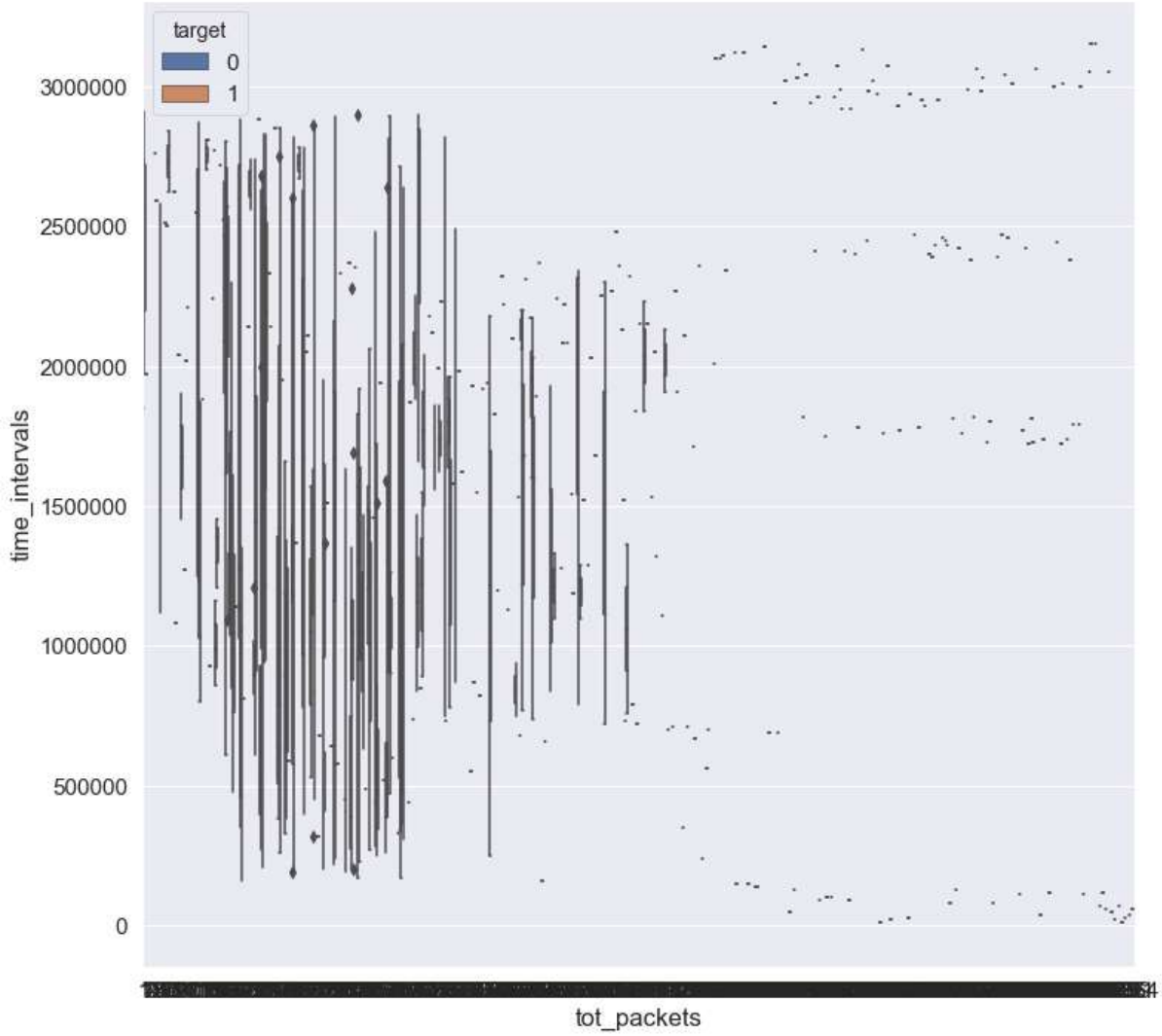

1996
2006
2014
2019
2033
2064
2067
2090
2094
2139
2160
2166
2213
2215
2216
2233
2244
2246
2280
2282
2290
2304
2305
2312
2319
2327
2340
2341
2349
2355
2363
2382
2383
2391
2392
2396
2397
2402
2415
2424
2434
2436
2438
2442
2446


2448
2450
2457
2458
2460
2461
2465
2466
2469
2473
2481
2490
2493
2496
2506
2513
2518
2524
2527
2531
2532
2544
2554
2560
2569
2604
2606
2608
2618
2620
2632
2658
2670
2702
2706
2734
2735
2745
2760
2784
2790
2793
2797
2802
2808


2810
2838
2874
2906
2924
2938
3030
3098
3196
3218
4223
4295
4421
4505
4999
5101
5564
3000000
2500000
2000000
1500000
1000000
500000
0
time_intervals
0
1
target

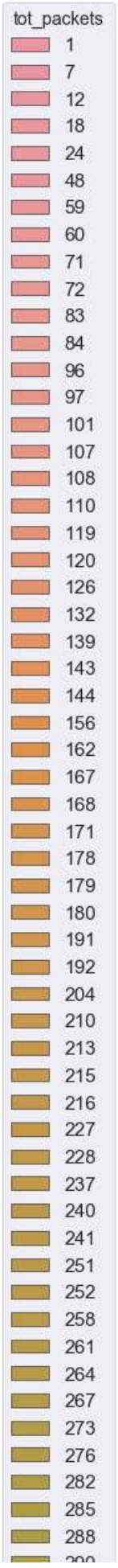
In [45]:

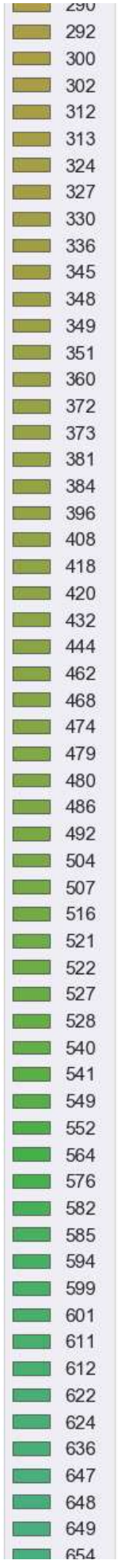
```
plt.figure(figsize=(12,12))
sns.boxplot(x='tot_packets',y='time_intervals',data=dff, hue='target')
plt.show()
```

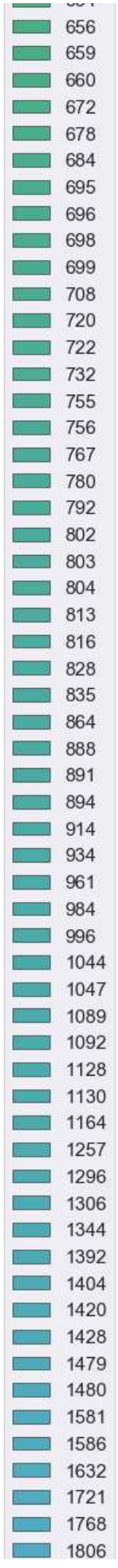

target
0
1
3000000
2500000
2000000
1500000
1000000
500000
0
time_intervals
tot_packets

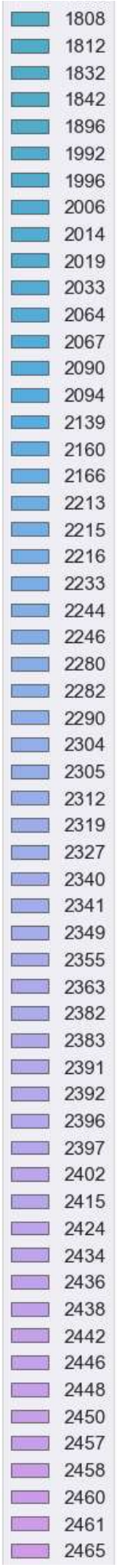
In [46]:

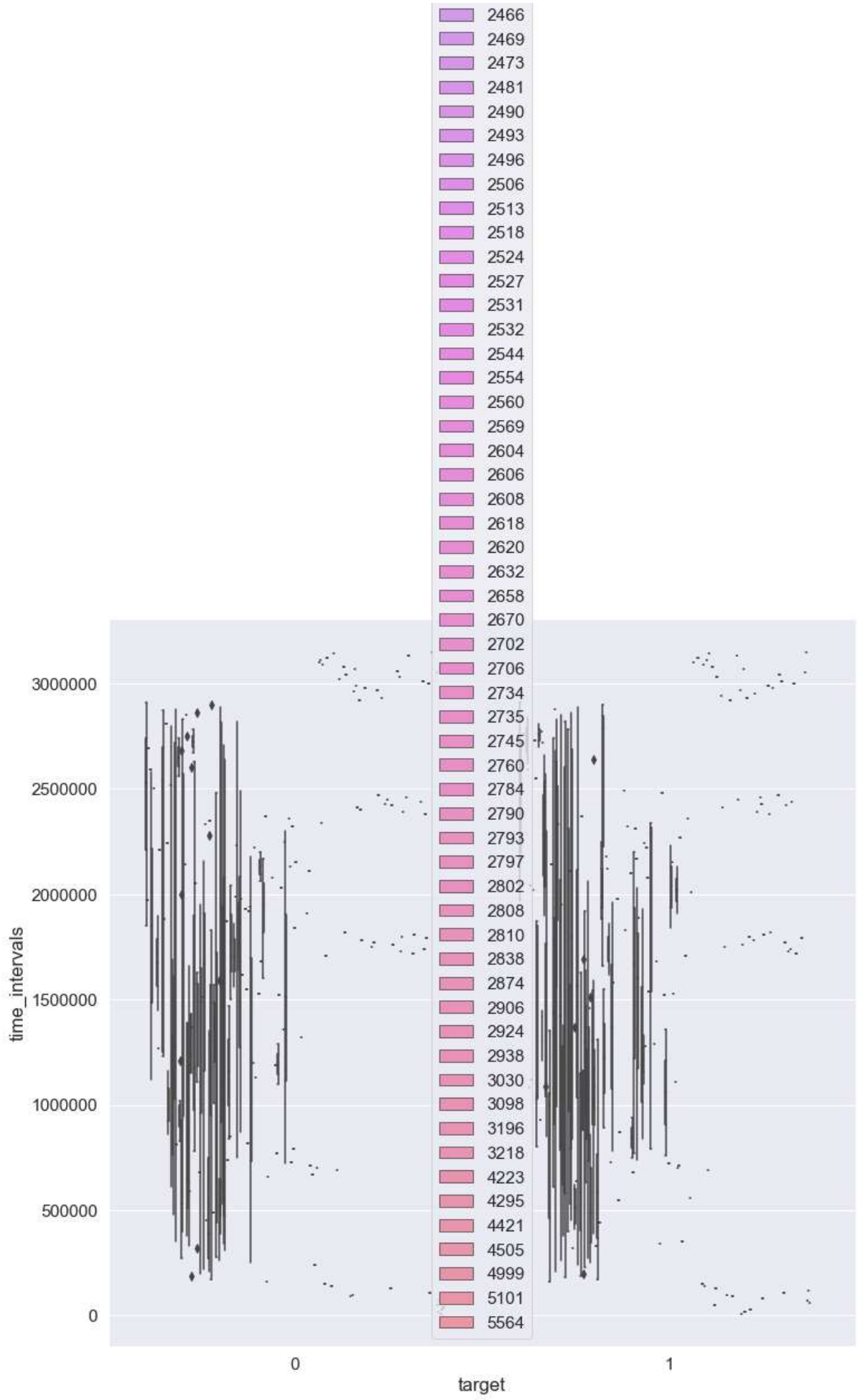
```
plt.figure(figsize=(12,12))
sns.boxplot(x='target',y='time_intervals',data=dff, hue='tot_packets')
plt.show()
```


tot_packets
1
7
12
18
24
48
59
60
71
72
83
84
96
97
101
107
108
110
119
120
126
132
139
143
144
156
162
167
168
171
178
179
180
191
192
204
210
213
215
216
227
228
237
240
241
251
252
258
261
264
267
273
276
282
285
288


290
292
300
302
312
313
324
327
330
336
345
348
349
351
360
372
373
381
384
396
408
418
420
432
444
462
468
474
479
480
486
492
504
507
516
521
522
527
528
540
541
549
552
564
576
582
585
594
599
601
611
612
622
624
636
647
648
649
654


656
659
660
672
678
684
695
696
698
699
708
720
722
732
755
756
767
780
792
802
803
804
813
816
828
835
864
888
891
894
914
934
961
984
996
1044
1047
1089
1092
1128
1130
1164
1257
1296
1306
1344
1392
1404
1420
1428
1479
1480
1581
1586
1632
1721
1768
1806


1808
1812
1832
1842
1896
1992
1996
2006
2014
2019
2033
2064
2067
2090
2094
2139
2160
2166
2213
2215
2216
2233
2244
2246
2280
2282
2290
2304
2305
2312
2319
2327
2340
2341
2349
2355
2363
2382
2383
2391
2392
2396
2397
2402
2415
2424
2434
2436
2438
2442
2446
2448
2450
2457
2458
2460
2461
2465


2466
2469
2473
2481
2490
2493
2496
2506
2513
2518
2524
2527
2531
2532
2544
2554
2560
2569
2604
2606
2608
2618
2620
2632
2658
2670
2702
2706
2734
2735
2745
2760
2784
2790
2793
2797
2802
2808
2810
2838
2874
2906
2924
2938
3030
3098
3196
3218
4223
4295
4421
4505
4999
5101
5564
3000000
2500000
2000000
1500000
1000000
500000
0
time_intervals
0
1
target