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-1000.csv')
```

In [3]:

```
dfm = pd.read_csv('wat-mal-time-interval-1000.csv')
```

In [4]:

```
dfg.head()
```

Out[4]:

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_north	inport_south	inport_eas
0	1000	65	20	6	4	46	27	22	19	21	...	69	100	14
1	2000	68	24	10	7	43	12	8	0	21	...	76	65	15
2	3000	47	10	9	1	31	1	2	2	31	...	65	65	13
3	4000	54	6	2	2	48	9	5	7	43	...	115	50	23
4	5000	60	12	3	1	36	13	12	6	26	...	65	55	13

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	inport_eas
0	1000	66	86	22	12	47	70	29	26	27	...	164	146	33
1	2000	77	91	15	10	51	61	10	4	28	...	172	109	37
2	3000	53	69	14	4	31	35	4	4	31	...	154	106	28
3	4000	62	55	9	2	50	43	6	7	45	...	190	63	40
4	5000	69	66	7	3	49	54	29	18	32	...	132	88	27

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	inport_eas
0	1000	65	20	6	4	46	27	22	19	21	...	69	100	14
1	2000	68	24	10	7	43	12	8	0	21	...	76	65	15
2	3000	47	10	9	1	31	1	2	2	31	...	65	65	13
3	4000	54	6	2	2	48	9	5	7	43	...	115	50	23
4	5000	60	12	3	1	36	13	12	6	26	...	65	55	13

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_south	inport
0	1000	65	20	6	4	46	27	22	19	21	...	69	100	
3152	1000	66	86	22	12	47	70	29	26	27	...	164	146	
3153	2000	77	91	15	10	51	61	10	4	28	...	172	109	
1	2000	68	24	10	7	43	12	8	0	21	...	76	65	
2	3000	47	10	9	1	31	1	2	2	31	...	65	65	
3154	3000	53	69	14	4	31	35	4	4	31	...	154	106	
3	4000	54	6	2	2	48	9	5	7	43	...	115	50	
3155	4000	62	55	9	2	50	43	6	7	45	...	190	63	
4	5000	60	12	3	1	36	13	12	6	26	...	65	55	
3156	5000	69	66	7	3	49	54	29	18	32	...	132	88	
3157	6000	69	80	19	10	56	42	8	9	44	...	163	104	
5	6000	61	11	5	3	49	5	7	8	38	...	85	70	
3158	7000	66	77	12	8	32	51	14	7	20	...	134	82	
6	7000	59	16	7	6	26	3	1	0	20	...	50	45	
3159	8000	61	95	34	15	39	40	6	8	34	...	131	102	
7	8000	60	22	10	3	38	11	6	8	33	...	77	71	
3160	9000	77	115	35	15	37	63	23	13	13	...	107	123	
8	9000	73	49	30	12	23	6	6	0	6	...	27	79	
3161	10000	75	91	34	16	53	62	50	21	26	...	132	153	
9	10000	66	24	23	9	45	23	36	20	25	...	66	111	
3162	11000	88	104	39	25	53	42	19	1	23	...	136	103	
10	11000	84	37	25	12	38	5	18	0	19	...	64	59	
3163	12000	79	110	77	45	62	28	14	3	39	...	134	172	
11	12000	72	29	28	15	44	6	13	3	27	...	81	92	
12	13000	71	38	31	12	33	17	24	0	5	...	41	82	
3164	13000	91	107	50	12	53	52	30	0	12	...	71	109	
13	14000	73	34	33	3	38	22	22	13	17	...	58	106	
3165	14000	83	125	75	23	44	48	31	15	11	...	115	163	
14	15000	77	44	31	12	34	23	6	6	24	...	46	64	
3166	15000	79	95	39	15	40	65	22	14	24	...	119	113	
3167	16000	80	76	28	20	35	44	14	8	17	...	112	104	
15	16000	84	37	24	19	33	12	10	7	10	...	42	87	
16	17000	48	12	5	1	38	17	21	6	16	...	61	57	
3168	17000	61	57	17	8	50	34	19	7	16	...	96	94	

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_north	inport_south	inport
17	18000	36	5	3	1	31	2	2	1	31	...	75	45	
3169	18000	37	42	7	3	32	33	14	8	37	...	121	66	
18	19000	42	20	13	7	16	6	0	1	5	...	25	50	
3170	19000	49	68	25	13	17	34	6	7	6	...	101	85	
19	20000	42	10	9	7	32	12	13	13	21	...	51	58	
3171	20000	60	76	27	7	33	26	13	13	22	...	67	76	
20	21000	42	9	2	1	21	1	1	1	21	...	60	30	
3172	21000	49	92	44	31	27	21	19	13	21	...	82	92	
3173	22000	67	106	51	31	35	56	19	24	5	...	111	103	
21	22000	60	37	31	18	23	0	1	6	5	...	26	46	
22	23000	73	20	7	0	41	18	13	6	24	...	52	46	
3174	23000	92	91	39	19	41	35	12	7	30	...	98	86	
3175	24000	98	92	27	3	69	65	33	18	36	...	135	105	
23	24000	77	22	15	2	55	32	33	12	30	...	98	85	
24	25000	73	38	25	12	17	18	0	0	0	...	16	46	
3176	25000	73	116	50	18	18	70	19	6	1	...	107	86	

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  outport_l
   0            1000  65  20   6   4  46  27  22  19  21  ...           14           20
3152            1000  66  86  22  12  47  70  29  26  27  ...           33           34
3153            2000  77  91  15  10  51  61  10   4  28  ...           37           44
   1            2000  68  24  10   7  43  12   8   0  21  ...           15           16
   2            3000  47  10   9   1  31   1   2   2  31  ...           13           10
 ...             ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...          ...          ...
6301         3150000  77  24  18  18  41  12   6  12  35  ...           54           42
3149         3150000  81  39  30  18  42  26  24  24  24  ...           70           87
6302         3151000  73  36  30  18  36  12  18   6  30  ...           66           66
3150         3151000  69  15   6   6  42  16   6  12  36  ...           62           27
3151         3152000  78  42  36  18  36  12  22   6  30  ...           77           78

6303 rows × 34 columns
```

In [13]:

```
df.to_csv('wat-time-interval-1000.csv',index=False)
```

In [14]:

```
df = pd.read_csv('wat-time-interval-1000.csv')
```

In [15]:

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

Out[15]:

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_east	inport_west	outport_lo
0	1000	65	20	6	4	46	27	22	19	21	...	14	20	
1	1000	66	86	22	12	47	70	29	26	27	...	33	34	
2	2000	77	91	15	10	51	61	10	4	28	...	37	44	
3	2000	68	24	10	7	43	12	8	0	21	...	15	16	
4	3000	47	10	9	1	31	1	2	2	31	...	13	10	
5	3000	53	69	14	4	31	35	4	4	31	...	28	23	
6	4000	54	6	2	2	48	9	5	7	43	...	23	10	
7	4000	62	55	9	2	50	43	6	7	45	...	40	22	
8	5000	60	12	3	1	36	13	12	6	26	...	13	11	
9	5000	69	66	7	3	49	54	29	18	32	...	27	42	
10	6000	69	80	19	10	56	42	8	9	44	...	30	36	
11	6000	61	11	5	3	49	5	7	8	38	...	17	14	
12	7000	66	77	12	8	32	51	14	7	20	...	27	25	
13	7000	59	16	7	6	26	3	1	0	20	...	10	9	
14	8000	61	95	34	15	39	40	6	8	34	...	50	36	
15	8000	60	22	10	3	38	11	6	8	33	...	25	25	
16	9000	77	115	35	15	37	63	23	13	13	...	31	57	
17	9000	73	49	30	12	23	6	6	0	6	...	15	29	
18	10000	75	91	34	16	53	62	50	21	26	...	48	74	
19	10000	66	24	23	9	45	23	36	20	25	...	34	56	
20	11000	88	104	39	25	53	42	19	1	23	...	68	72	
21	11000	84	37	25	12	38	5	18	0	19	...	40	50	
22	12000	79	110	77	45	62	28	14	3	39	...	70	80	
23	12000	72	29	28	15	44	6	13	3	27	...	48	54	
24	13000	71	38	31	12	33	17	24	0	5	...	22	36	
25	13000	91	107	50	12	53	52	30	0	12	...	44	60	
26	14000	73	34	33	3	38	22	22	13	17	...	14	38	
27	14000	83	125	75	23	44	48	31	15	11	...	51	85	
28	15000	77	44	31	12	34	23	6	6	24	...	44	50	
29	15000	79	95	39	15	40	65	22	14	24	...	68	64	
30	16000	80	76	28	20	35	44	14	8	17	...	32	58	
31	16000	84	37	24	19	33	12	10	7	10	...	15	41	
32	17000	48	12	5	1	38	17	21	6	16	...	14	13	
33	17000	61	57	17	8	50	34	19	7	16	...	25	27	

	time_intervals	r0	r1	r2	r3	r4	r5	r6	r7	r8	...	inport_east	inport_west	outport_lo
34	18000	36	5	3	1	31	2	2	1	31	...	15	9	
35	18000	37	42	7	3	32	33	14	8	37	...	25	20	
36	19000	42	20	13	7	16	6	0	1	5	...	5	10	
37	19000	49	68	25	13	17	34	6	7	6	...	46	33	
38	20000	42	10	9	7	32	12	13	13	21	...	15	26	
39	20000	60	76	27	7	33	26	13	13	22	...	24	49	
40	21000	42	9	2	1	21	1	1	1	21	...	12	12	
41	21000	49	92	44	31	27	21	19	13	21	...	25	52	
42	22000	67	106	51	31	35	56	19	24	5	...	71	83	
43	22000	60	37	31	18	23	0	1	6	5	...	16	50	
44	23000	73	20	7	0	41	18	13	6	24	...	39	22	
45	23000	92	91	39	19	41	35	12	7	30	...	64	72	
46	24000	98	92	27	3	69	65	33	18	36	...	73	67	
47	24000	77	22	15	2	55	32	33	12	30	...	57	39	
48	25000	73	38	25	12	17	18	0	0	0	...	8	40	
49	25000	73	116	50	18	18	70	19	6	1	...	79	72	

50 rows × 34 columns

```
In [16]:
```

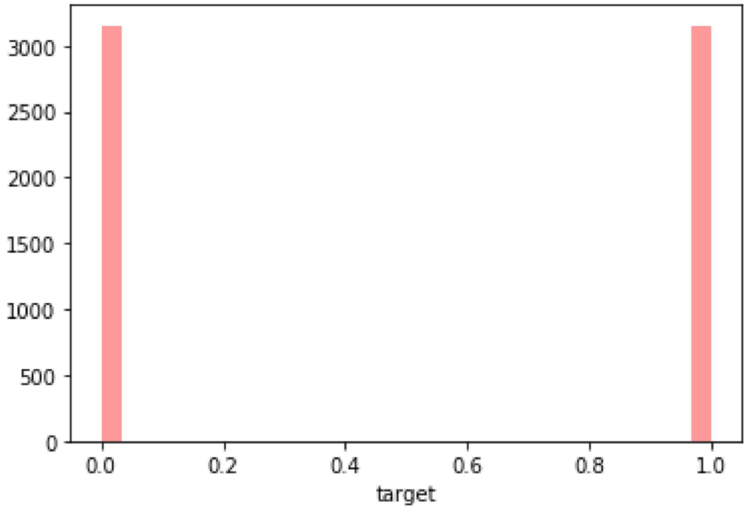
```python
import seaborn as sns
```

In [17]:

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

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x10ae967e248>
```



In [18]:

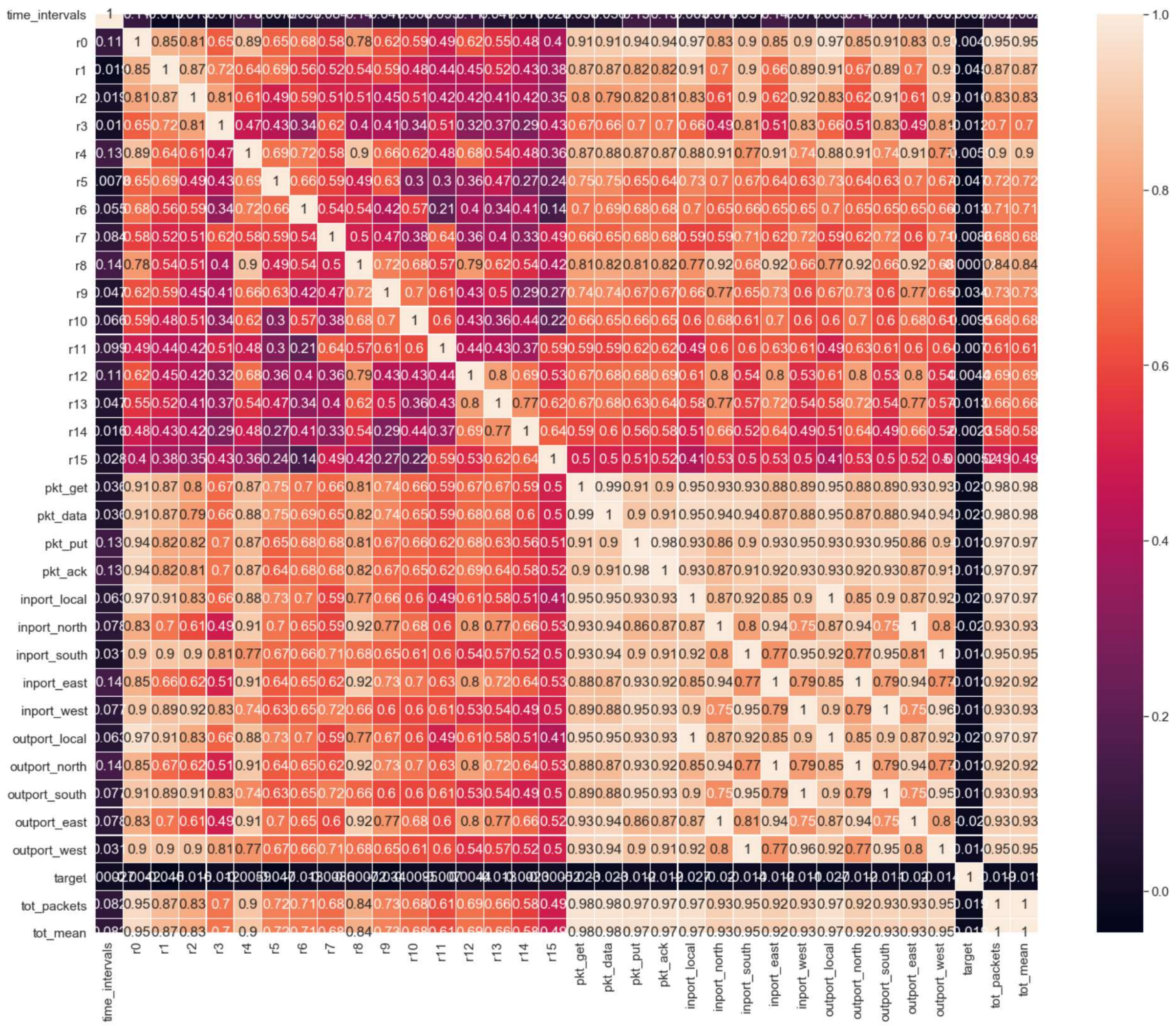
```
df.isnull().sum()
```

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	0.110224	-0.019140	0.018825	0.010450	0.129934	0.007815	0.0
r0	0.110224	1.000000	0.852018	0.813404	0.646446	0.888406	0.654695	0.6
r1	-0.019140	0.852018	1.000000	0.873634	0.723998	0.644690	0.685840	0.5
r2	0.018825	0.813404	0.873634	1.000000	0.811143	0.612568	0.491640	0.5
r3	0.010450	0.646446	0.723998	0.811143	1.000000	0.471697	0.432813	0.3
r4	0.129934	0.888406	0.644690	0.612568	0.471697	1.000000	0.693032	0.7
r5	0.007815	0.654695	0.685840	0.491640	0.432813	0.693032	1.000000	0.6
r6	0.055179	0.679588	0.564432	0.586711	0.337859	0.717216	0.664308	1.0
r7	0.084233	0.584164	0.521348	0.511068	0.615132	0.581269	0.587628	0.5
r8	0.141078	0.782763	0.542039	0.514305	0.401173	0.897094	0.493682	0.5
r9	0.047256	0.617532	0.592191	0.451041	0.408251	0.656349	0.625944	0.4
r10	0.065532	0.590443	0.483425	0.513602	0.335397	0.620614	0.304694	0.5
r11	0.099359	0.488456	0.435956	0.415162	0.510764	0.482547	0.302953	0.2
r12	0.111450	0.622318	0.446197	0.415544	0.316622	0.679061	0.356530	0.3
r13	0.047216	0.550734	0.516775	0.408336	0.366959	0.541399	0.466598	0.3
r14	0.015741	0.482100	0.434187	0.415857	0.287473	0.480264	0.270893	0.4
r15	0.028107	0.404000	0.381611	0.349589	0.426415	0.356925	0.242108	0.1
pkt_get	0.035693	0.911264	0.871099	0.795793	0.671809	0.870101	0.748074	0.6
pkt_data	0.035742	0.913187	0.865286	0.787608	0.664987	0.878399	0.745534	0.6
pkt_put	0.129560	0.942025	0.824803	0.820122	0.701583	0.865698	0.648290	0.6
pkt_ack	0.128733	0.940803	0.815239	0.811896	0.697444	0.873775	0.642601	0.6
inport_local	0.063283	0.974349	0.913271	0.827226	0.655528	0.880717	0.732460	0.6
inport_north	0.078012	0.833455	0.699022	0.607399	0.488633	0.911157	0.696336	0.6
inport_south	0.031100	0.898754	0.899875	0.895105	0.811946	0.767514	0.674658	0.6
inport_east	0.142564	0.847267	0.664602	0.620687	0.505624	0.905884	0.636526	0.6
inport_west	0.076559	0.904626	0.886241	0.915244	0.833711	0.742168	0.625827	0.6
outport_local	0.063065	0.974841	0.912469	0.826627	0.655636	0.882161	0.731828	0.6
outport_north	0.142849	0.847766	0.665922	0.621955	0.506490	0.905030	0.637258	0.6
outport_south	0.076888	0.905310	0.886324	0.914432	0.832323	0.742644	0.626732	0.6
outport_east	0.078017	0.833451	0.699477	0.608194	0.489444	0.911574	0.697044	0.6
outport_west	0.031097	0.898482	0.900332	0.896239	0.813263	0.766928	0.674018	0.6
target	0.000275	-0.004230	-0.044867	-0.015658	-0.012170	-0.005857	-0.047133	-0.0
tot_packets	0.082014	0.954266	0.871225	0.827053	0.698062	0.899537	0.723469	0.7
tot_mean	0.082014	0.954266	0.871225	0.827053	0.698062	0.899537	0.723469	0.7

34 rows × 34 columns

In [21]:

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

Out[21]:

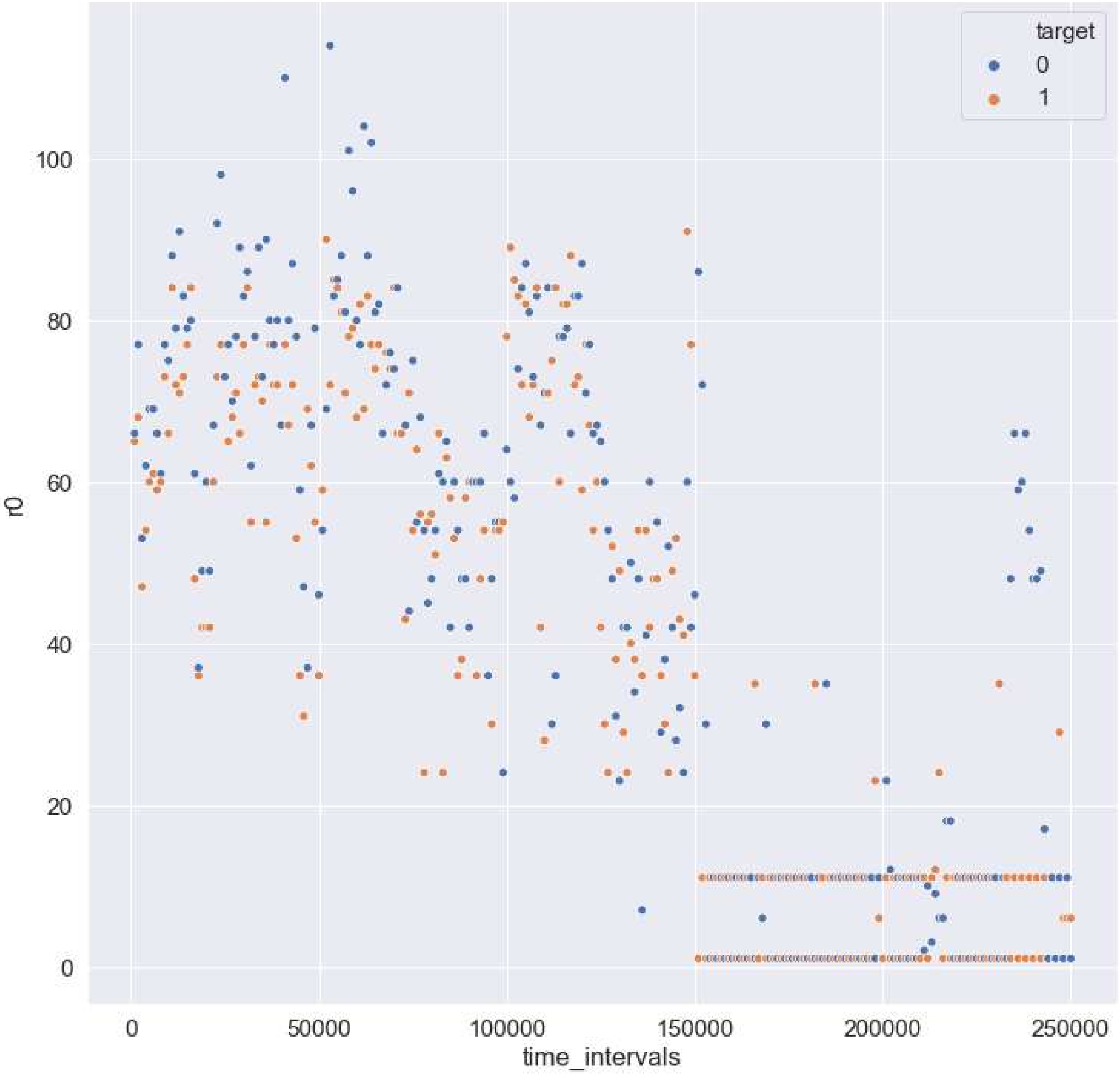
```
time_intervals    0.000275
r0               -0.004230
r1               -0.044867
r2               -0.015658
r3               -0.012170
r4               -0.005857
r5               -0.047133
r6               -0.013411
r7               -0.008642
r8               -0.000724
r9               -0.034100
r10              -0.009480
r11              -0.007039
r12               0.004404
r13              -0.012644
r14              -0.002295
r15              -0.000516
pkt_get          -0.023265
pkt_data         -0.023362
pkt_put          -0.011660
pkt_ack          -0.011682
inport_local     -0.026651
inport_north     -0.020379
inport_south     -0.013906
inport_east      -0.011612
inport_west      -0.011441
outport_local    -0.026574
outport_north    -0.011610
outport_south    -0.011498
outport_east     -0.020381
outport_west     -0.013906
target            1.000000
tot_packets      -0.018716
tot_mean         -0.018716
Name: target, dtype: float64
```

In [22]:

```
dff = pd.read_csv('wat-time-interval-1000.csv',nrows=500)
```

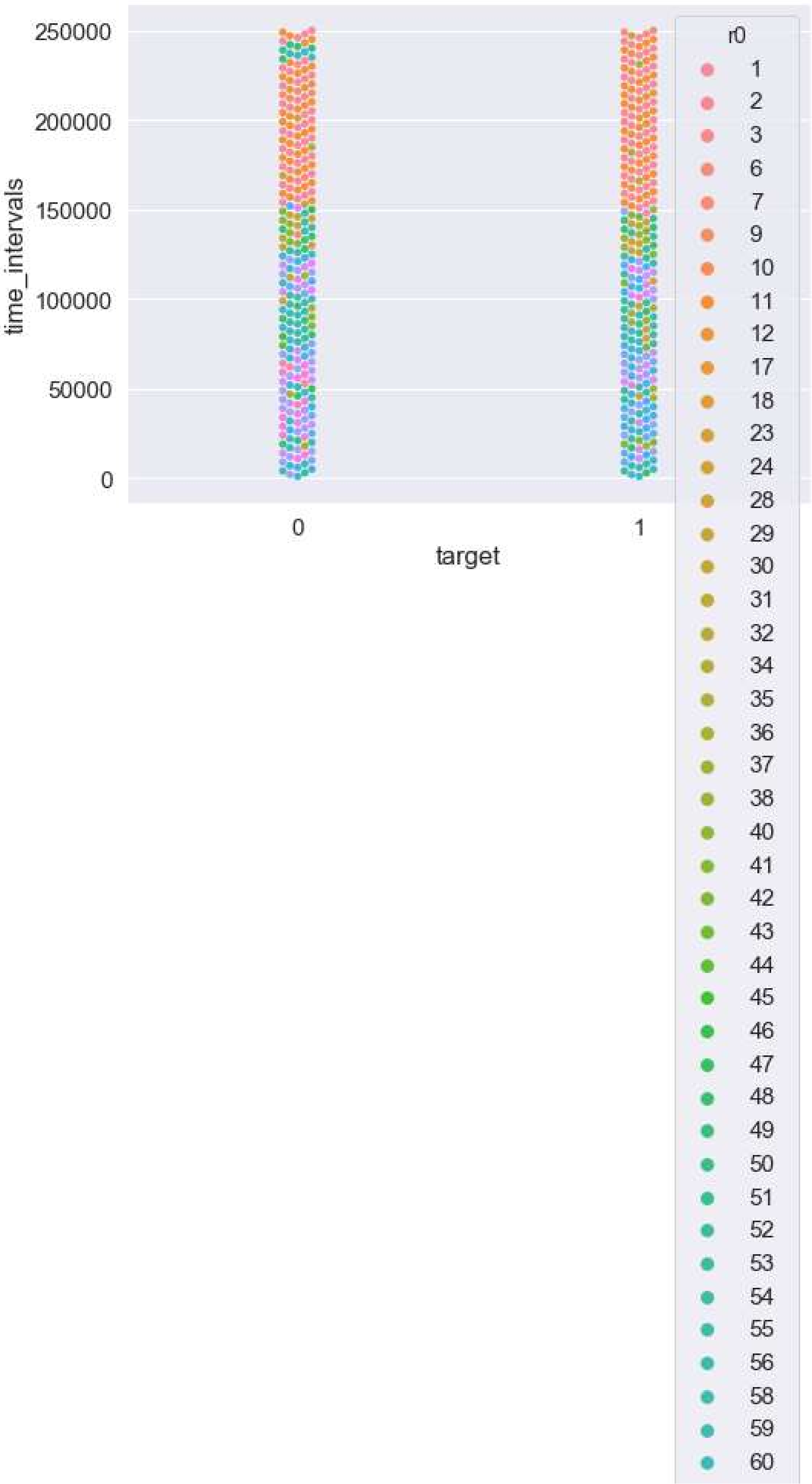
In [23]:

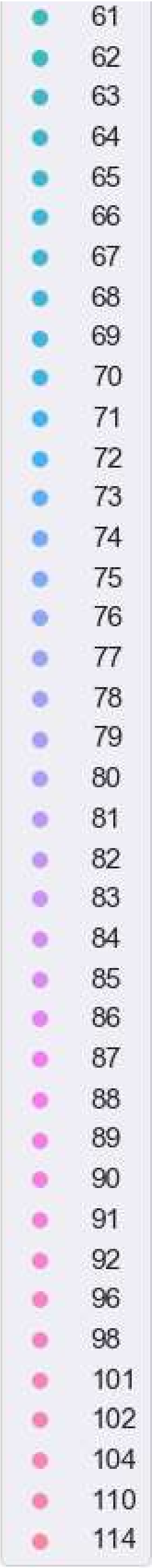
```
plt.figure(figsize=(12,12))
sns.scatterplot(x='time_intervals',y='r0',data=dff, hue='target')
plt.show()
```



In [24]:

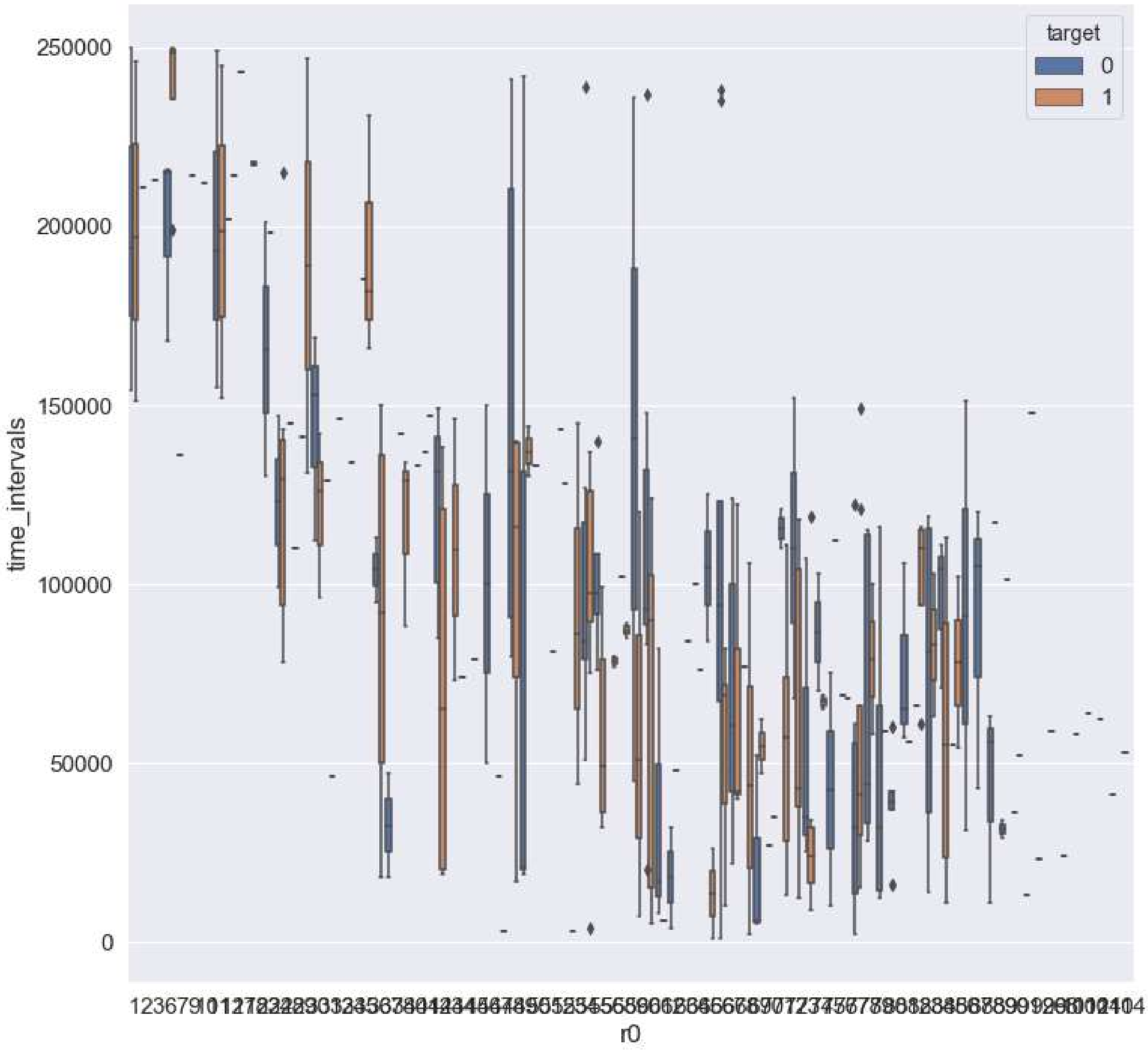
```
plt.figure(figsize=(8,6))
sns.swarmplot(x='target',y='time_intervals',data=dff, hue='r0')
plt.show()
```


250000
200000
150000
100000
50000
0
time_intervals
0
1
target
r0
1
2
3
6
7
9
10
11
12
17
18
23
24
28
29
30
31
32
34
35
36
37
38
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
58
59
60


● 61
● 62
● 63
● 64
● 65
● 66
● 67
● 68
● 69
● 70
● 71
● 72
● 73
● 74
● 75
● 76
● 77
● 78
● 79
● 80
● 81
● 82
● 83
● 84
● 85
● 86
● 87
● 88
● 89
● 90
● 91
● 92
● 96
● 98
● 101
● 102
● 104
● 110
● 114

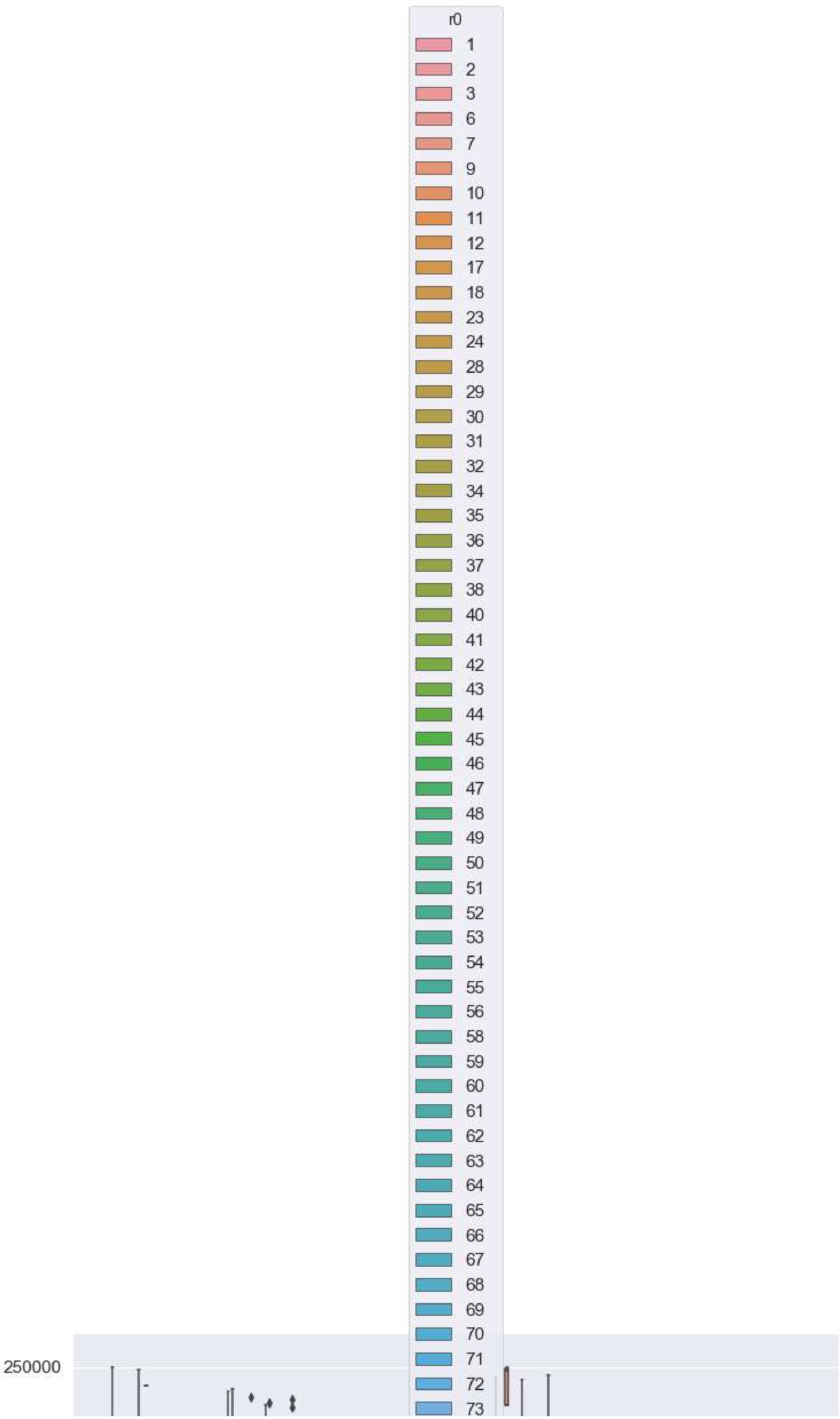
In [25]:

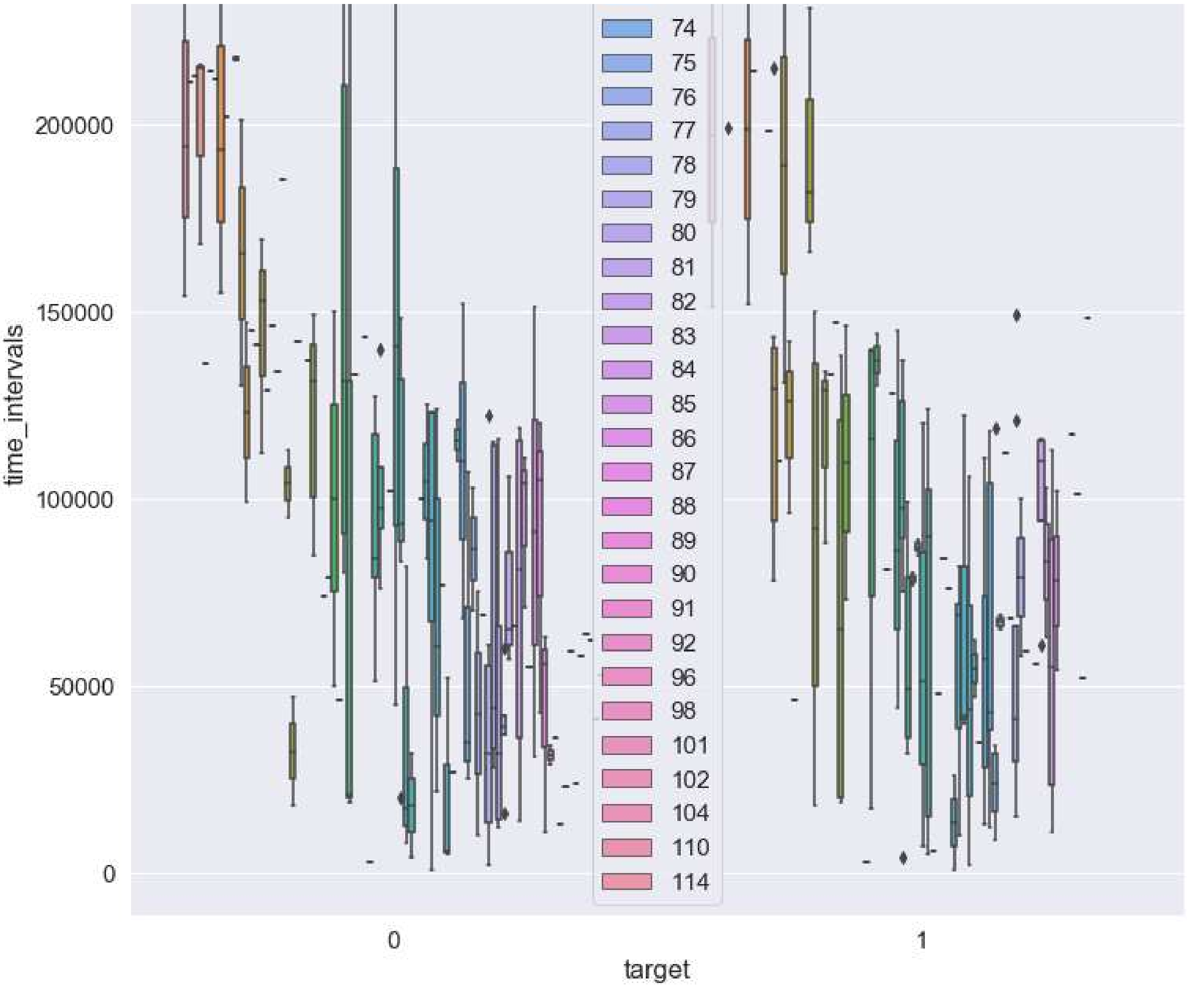
```
plt.figure(figsize=(12,12))
sns.boxplot(x='r0',y='time_intervals',data=dff, hue='target')
plt.show()
```



In [26]:

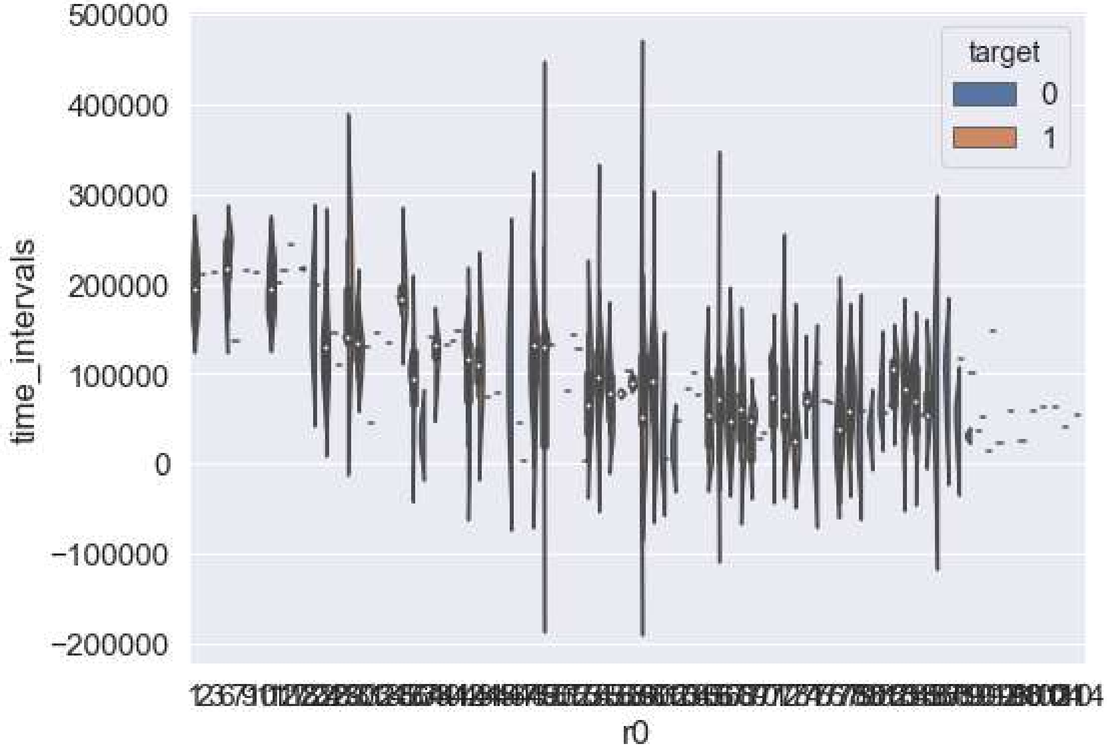
```
plt.figure(figsize=(12,12))
sns.boxplot(x='target',y='time_intervals',data=dff, hue='r0')
plt.show()
```


r0
1
2
3
6
7
9
10
11
12
17
18
23
24
28
29
30
31
32
34
35
36
37
38
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
250000


time_intervals
200000
150000
100000
50000
0
0
1
target
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
96
98
101
102
104
110
114

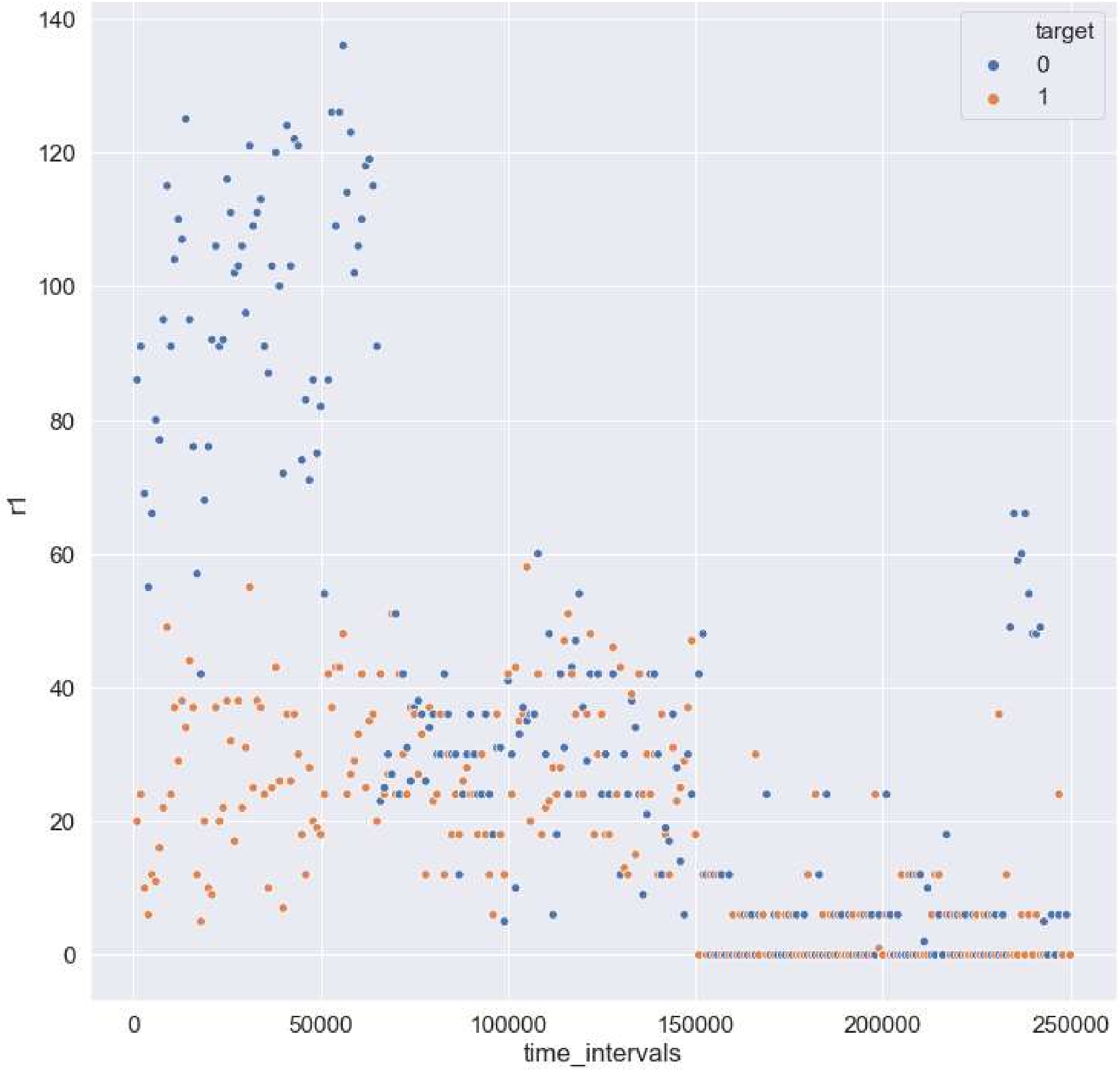
In [27]:

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



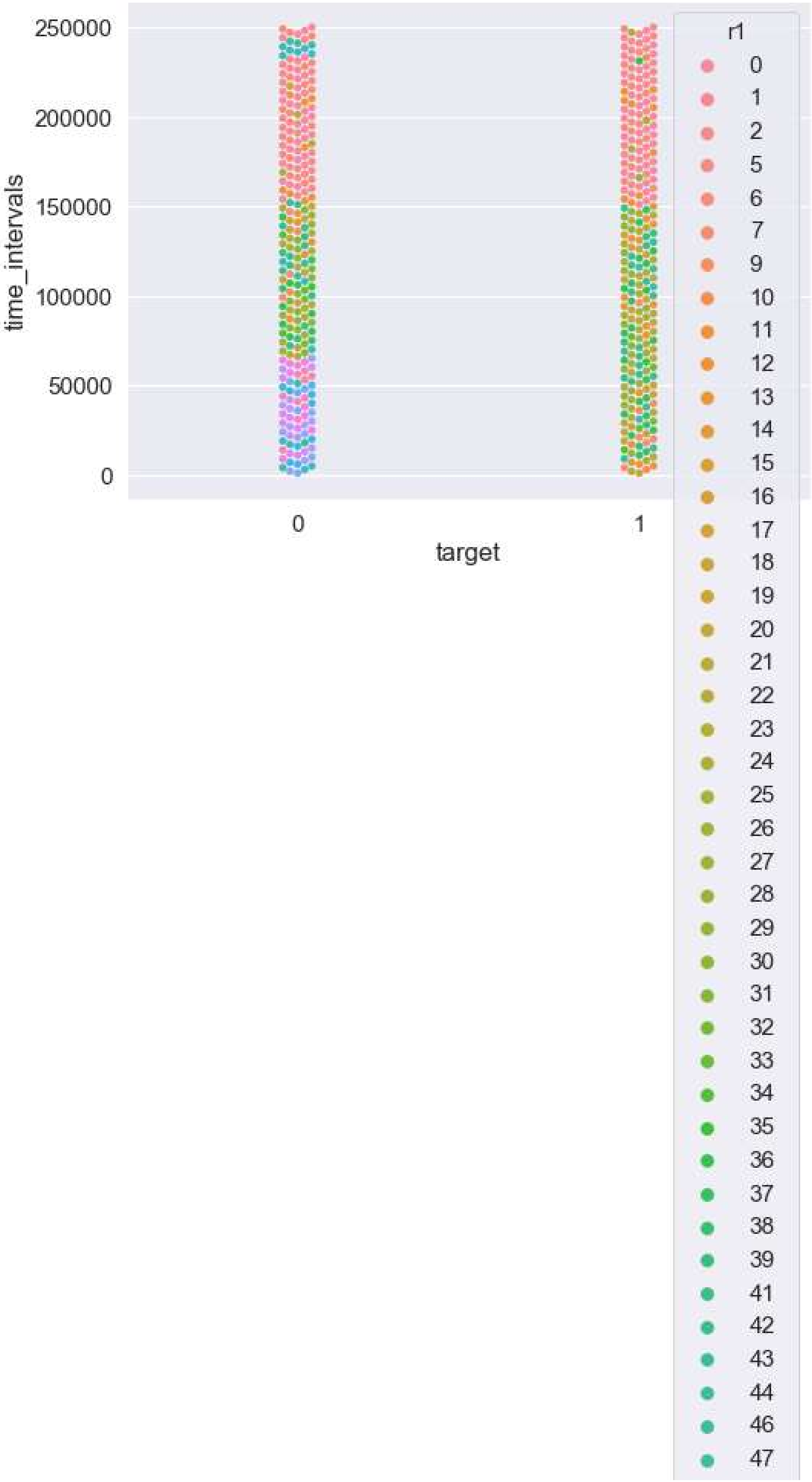
In [28]:

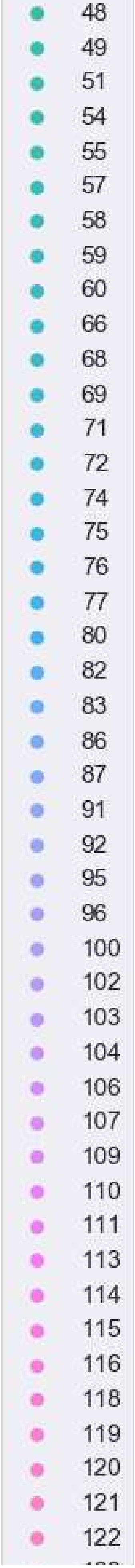
```
plt.figure(figsize=(12,12))
sns.scatterplot(x='time_intervals',y='r1',data=dff, hue='target')
plt.show()
```



In [29]:

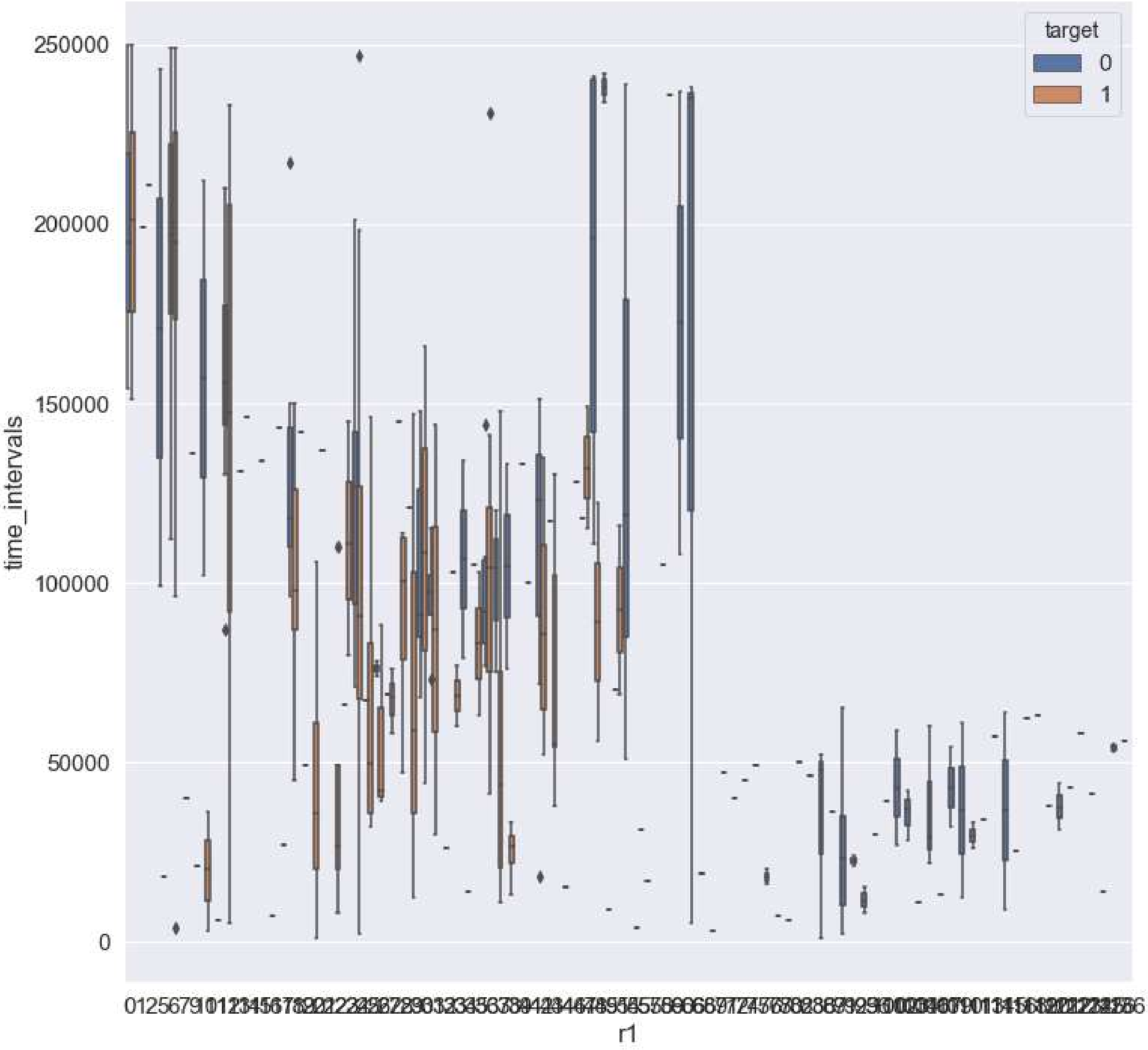
```
plt.figure(figsize=(8,6))
sns.swarmplot(x='target',y='time_intervals',data=dff, hue='r1')
plt.show()
```


time_intervals
250000
200000
150000
100000
50000
0
0
1
target
r1
0
1
2
5
6
7
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
41
42
43
44
46
47


48
49
51
54
55
57
58
59
60
66
68
69
71
72
74
75
76
77
80
82
83
86
87
91
92
95
96
100
102
103
104
106
107
109
110
111
113
114
115
116
118
119
120
121
122

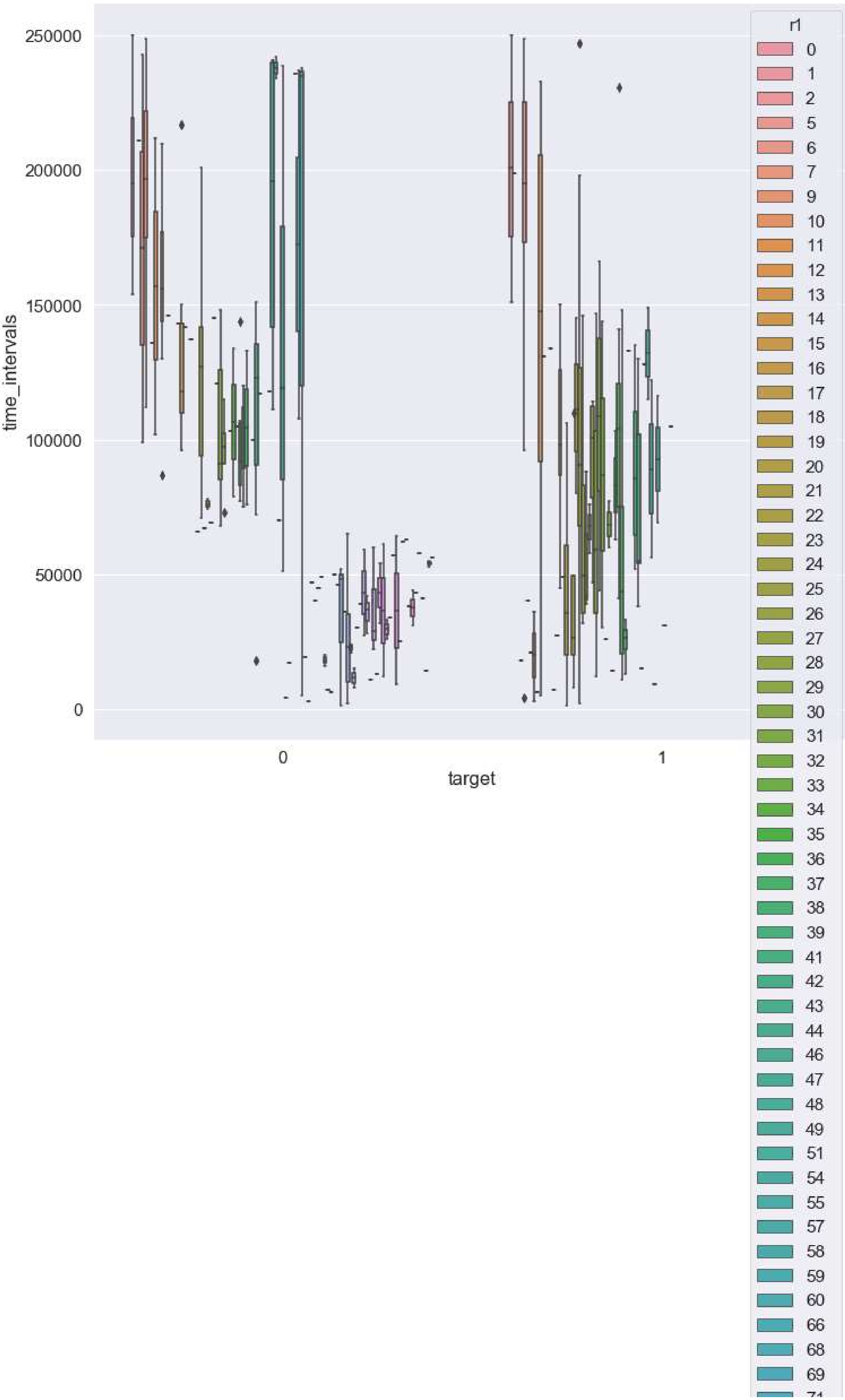
In [30]:

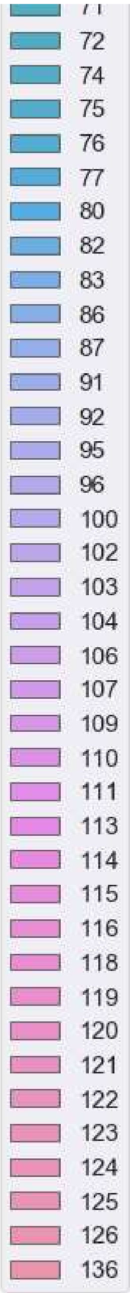
```
plt.figure(figsize=(12,12))
sns.boxplot(x='r1',y='time_intervals',data=dff, hue='target')
plt.show()
```



In [31]:

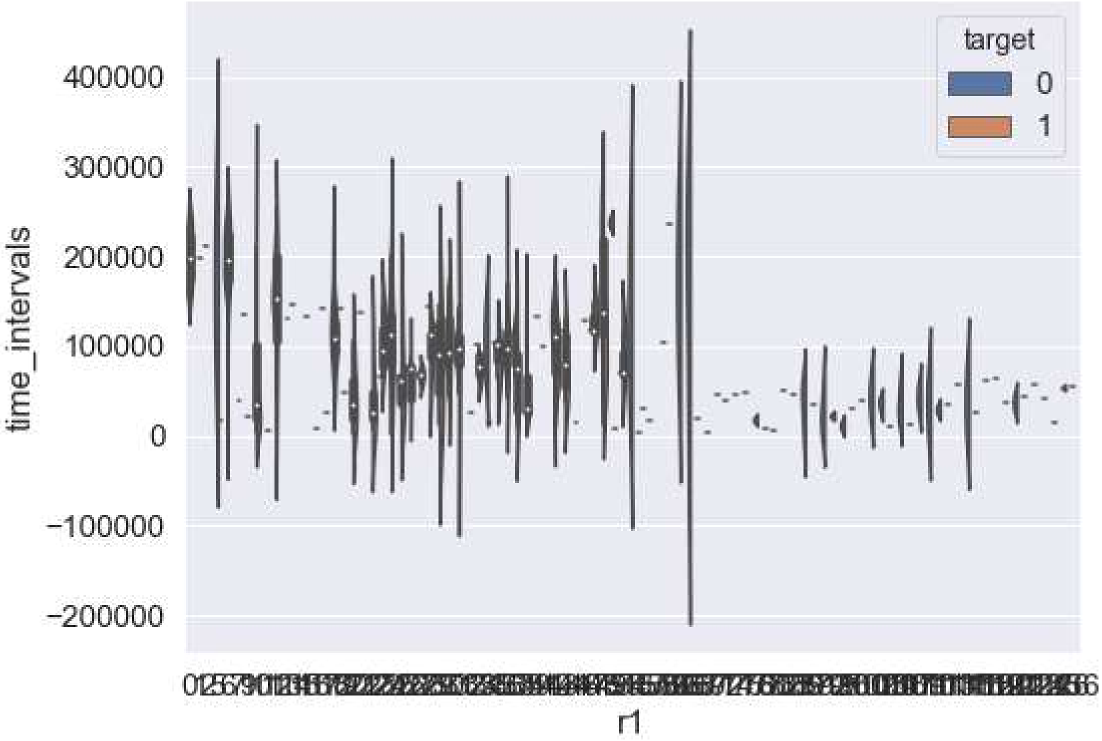
```
plt.figure(figsize=(12,12))
sns.boxplot(x='target',y='time_intervals',data=dff, hue='r1')
plt.show()
```


time_intervals
250000
200000
150000
100000
50000
0
0
1
target
r1
0
1
2
5
6
7
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
41
42
43
44
46
47
48
49
51
54
55
57
58
59
60
66
68
69
71


71
72
74
75
76
77
80
82
83
86
87
91
92
95
96
100
102
103
104
106
107
109
110
111
113
114
115
116
118
119
120
121
122
123
124
125
126
136

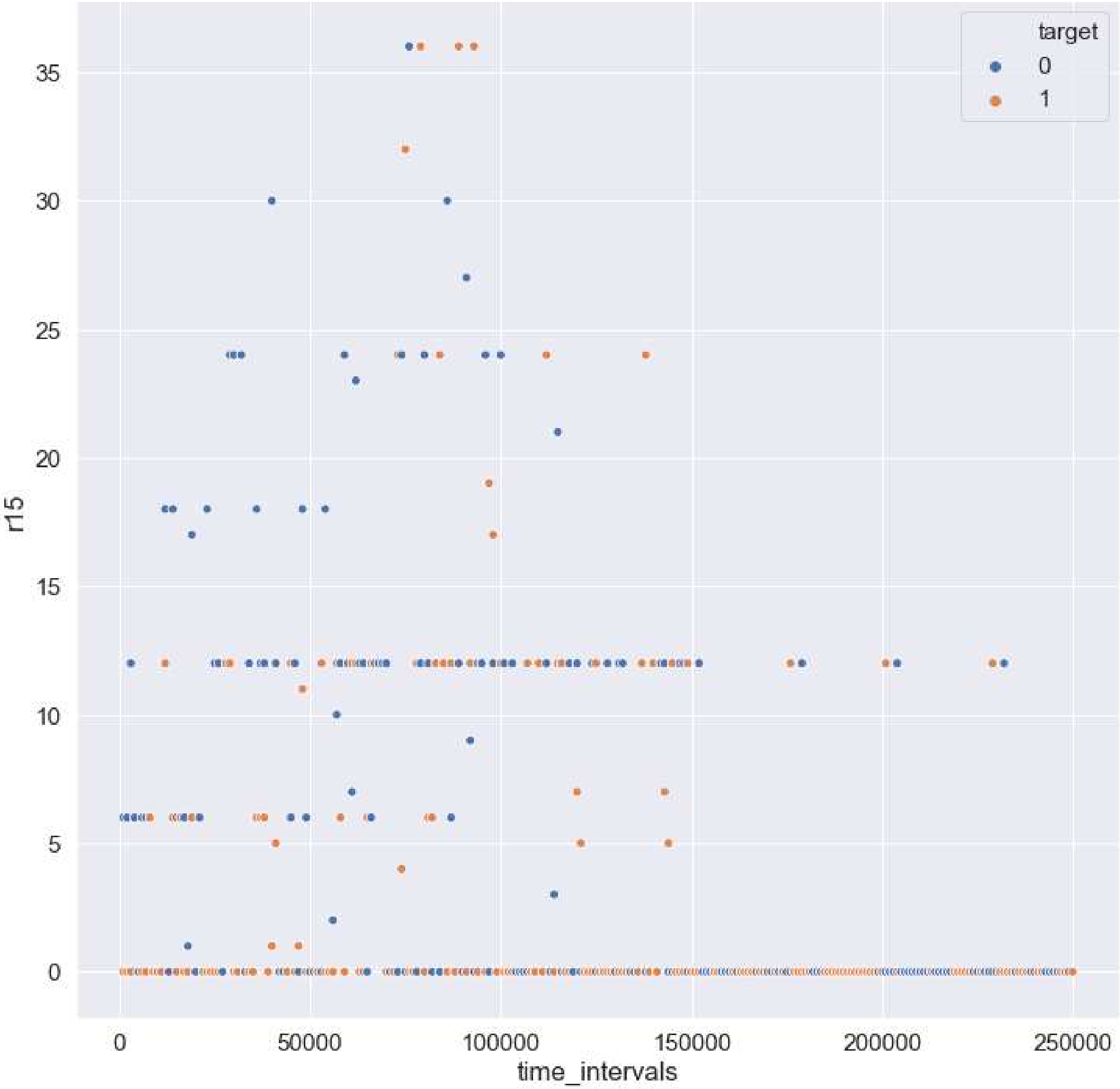
In [32]:

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



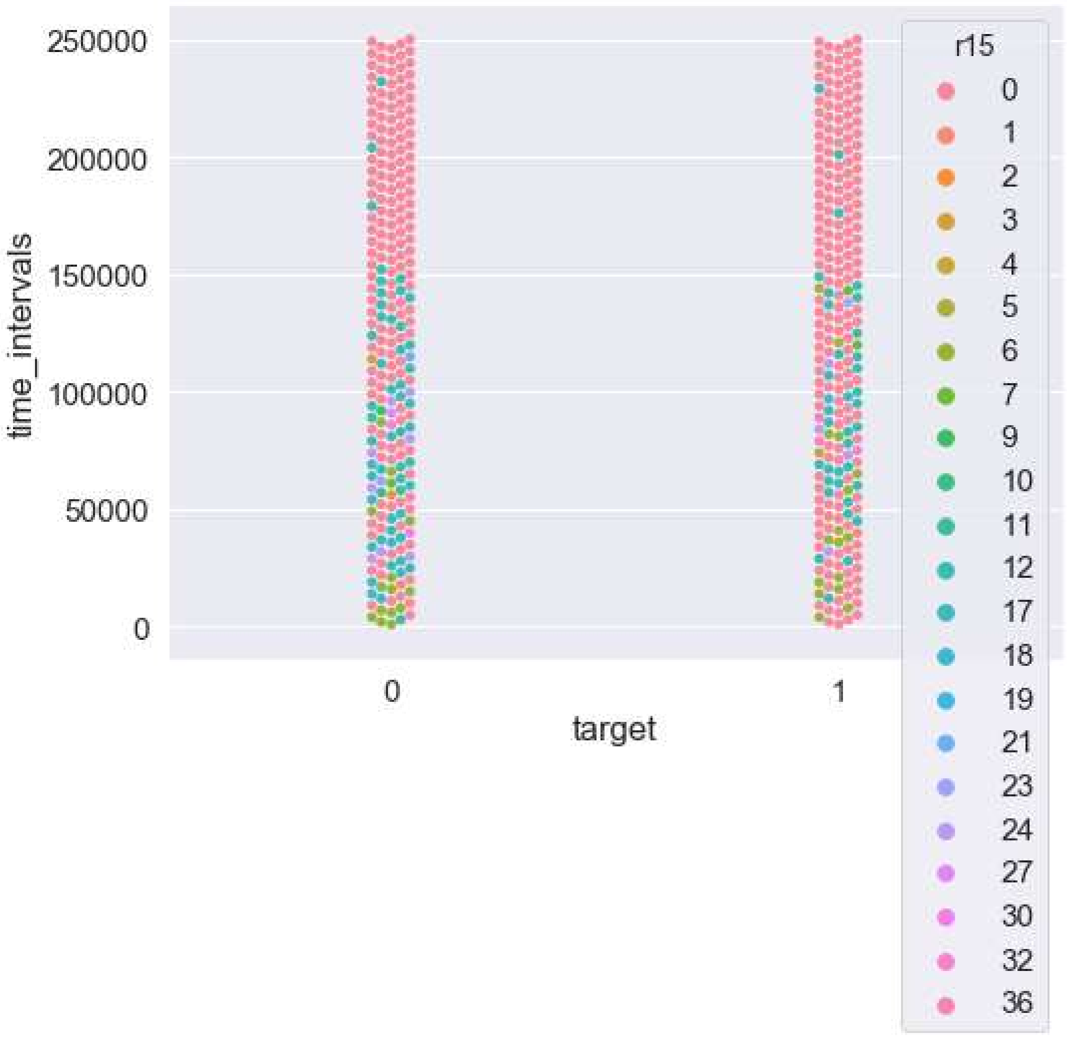
In [33]:

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



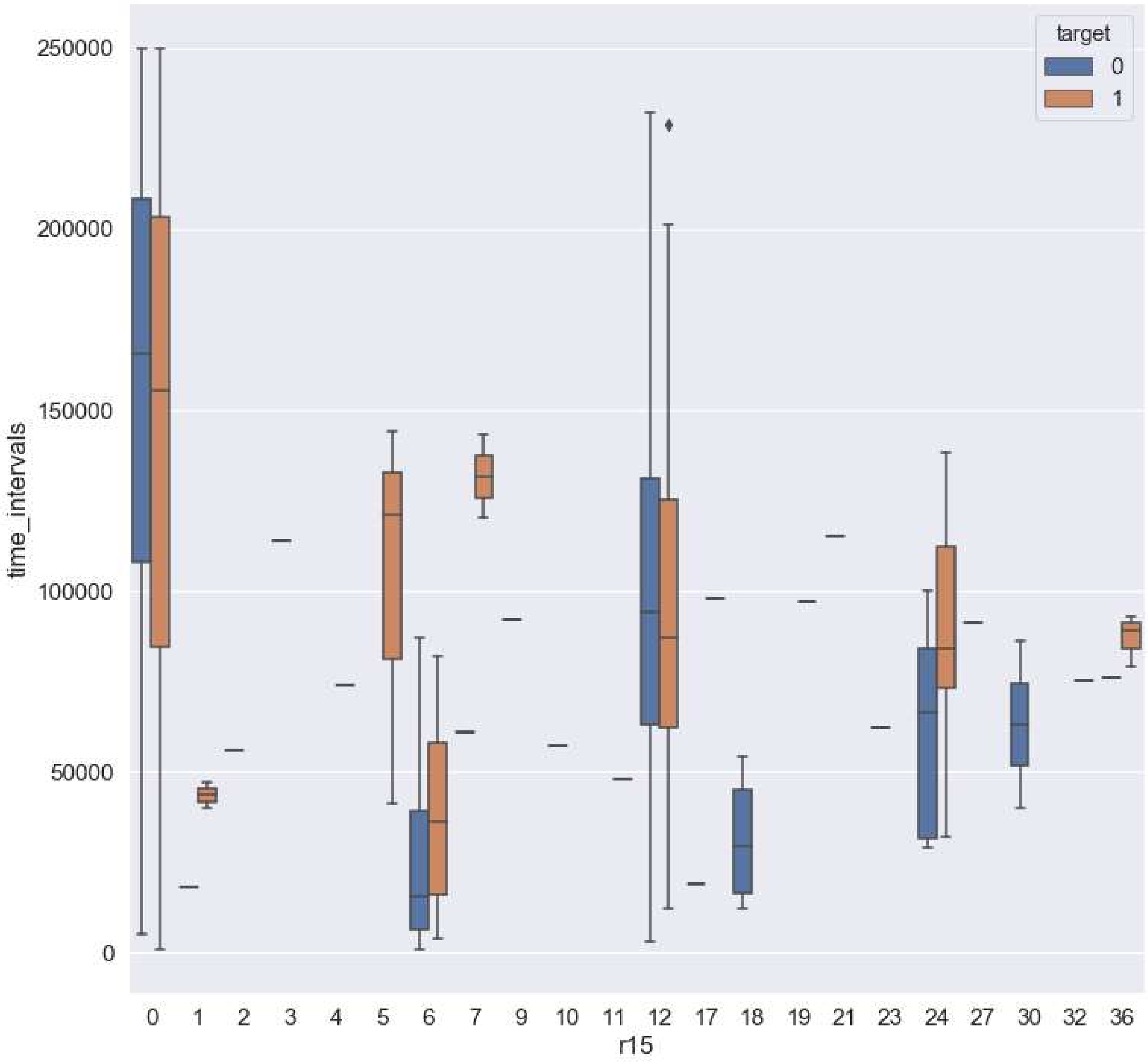
In [34]:

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



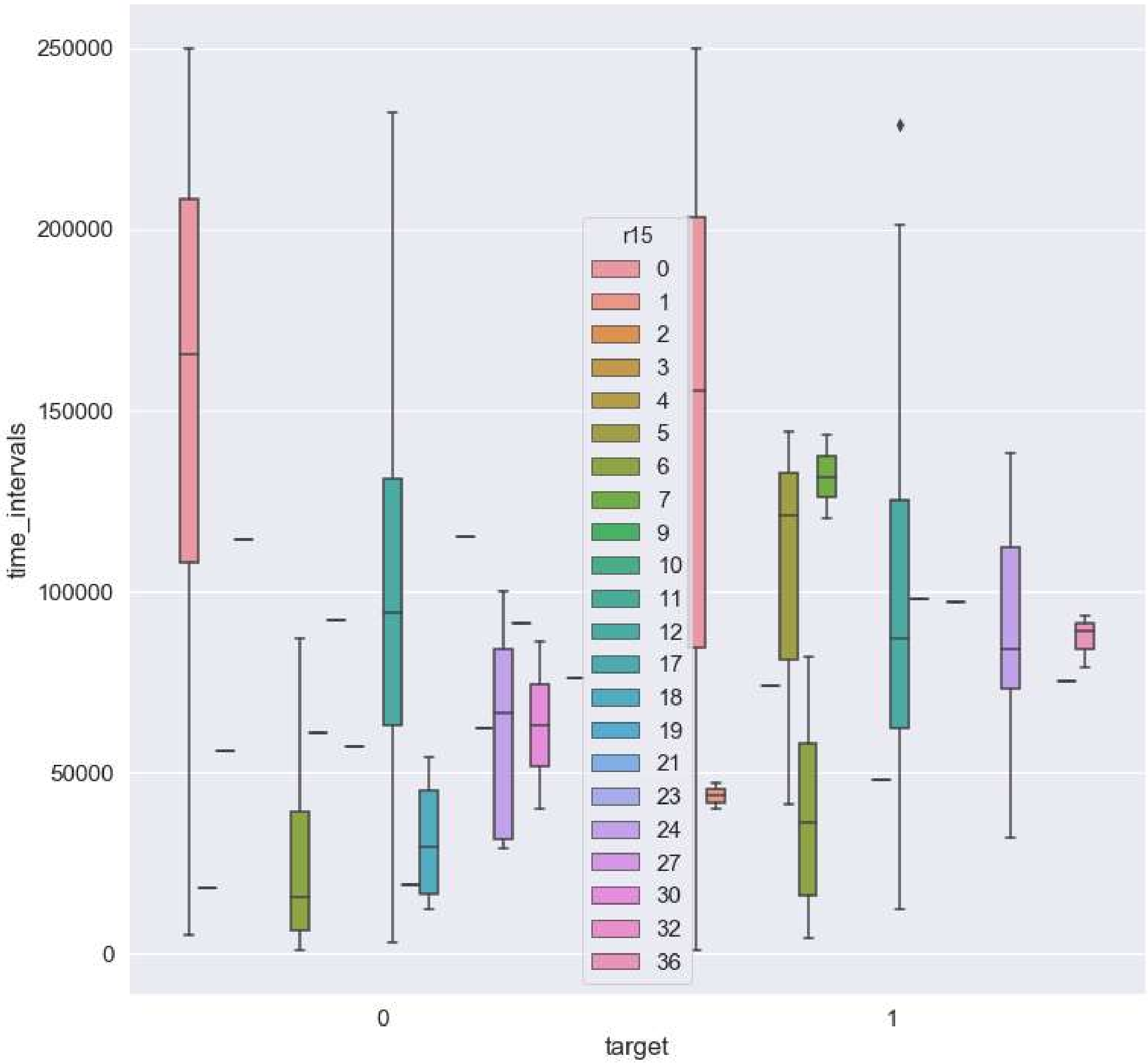
In [35]:

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



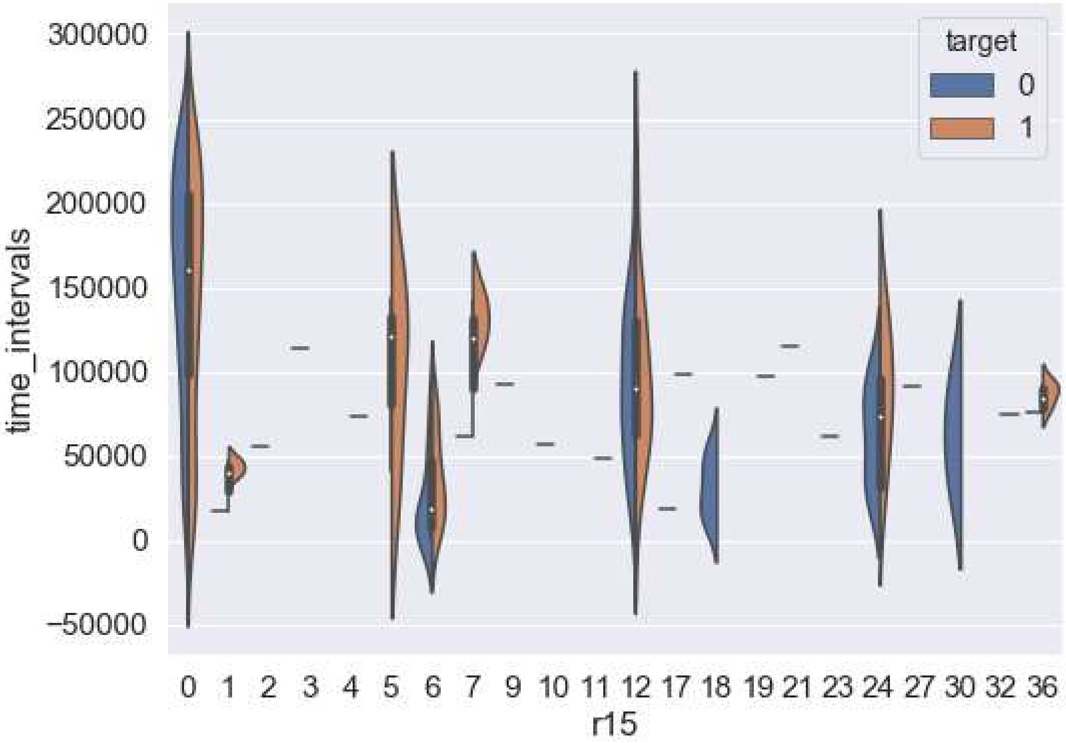
In [36]:

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



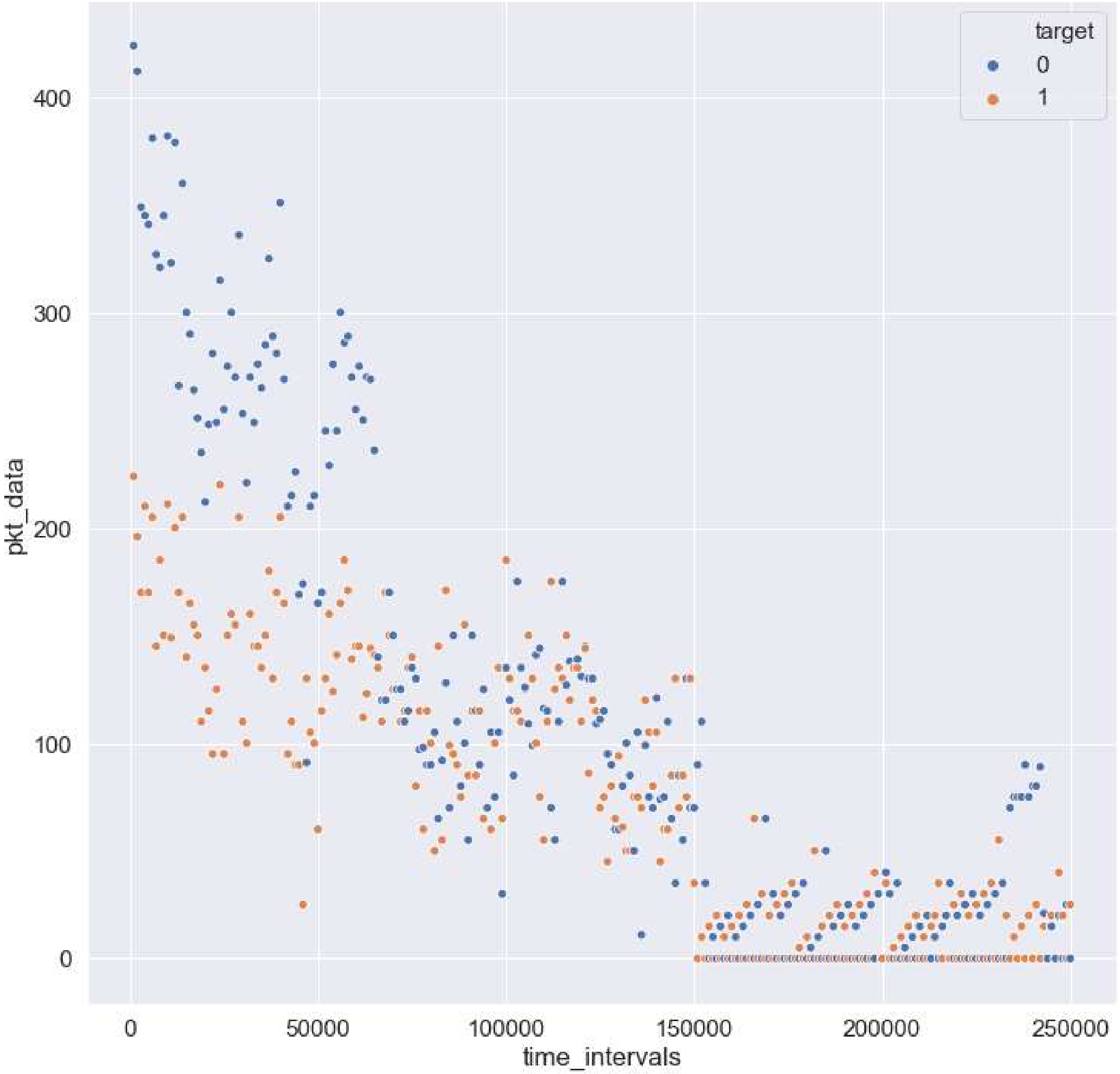
In [37]:

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



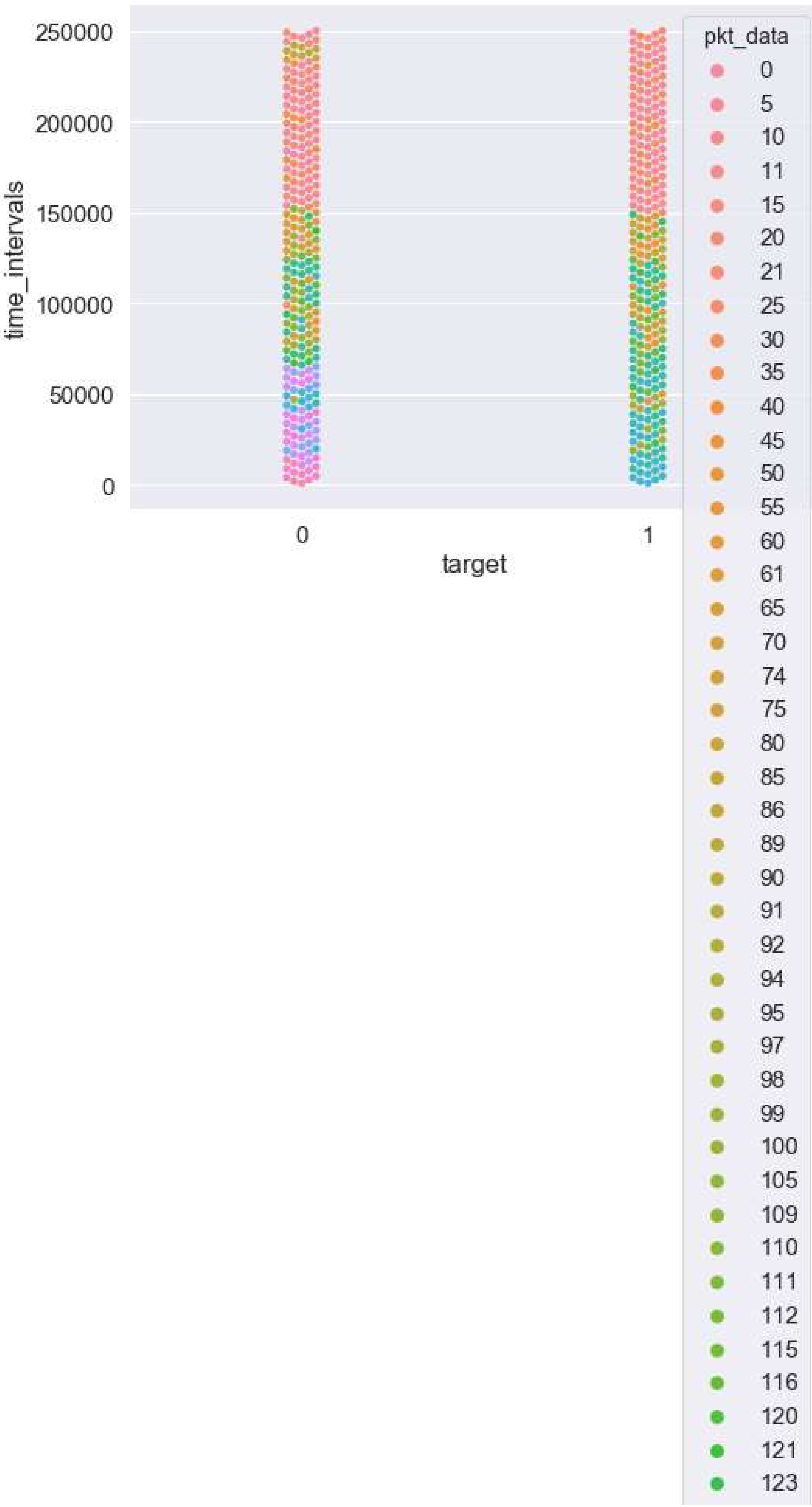
In [38]:

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

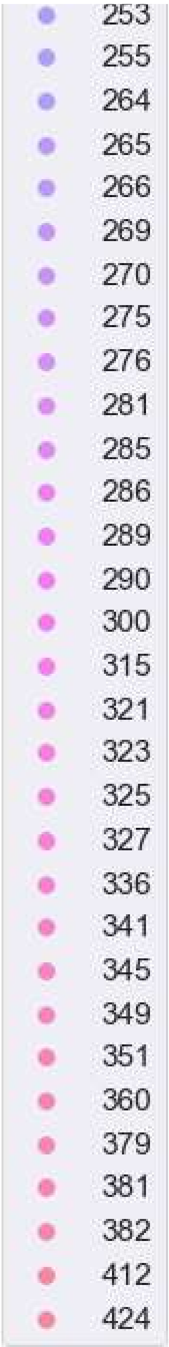


In [39]:

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

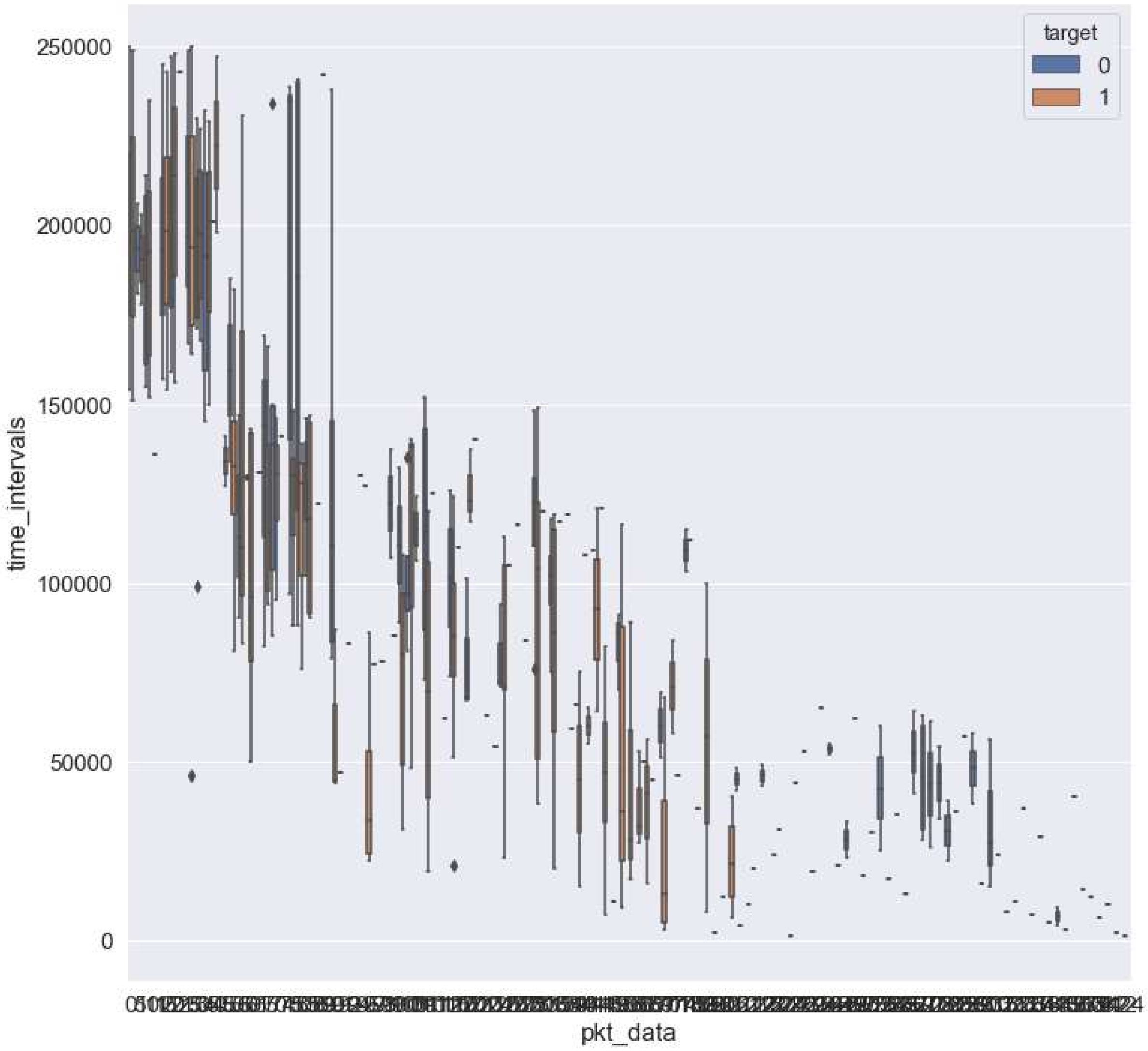

time_intervals
250000
200000
150000
100000
50000
0
0
1
target
pkt_data
0
5
10
11
15
20
21
25
30
35
40
45
50
55
60
61
65
70
74
75
80
85
86
89
90
91
92
94
95
97
98
99
100
105
109
110
111
112
115
116
120
121
123

124
125
126
127
128
130
131
135
138
139
140
141
144
145
149
150
155
160
165
169
170
171
174
175
180
185
196
200
205
210
211
212
215
220
221
224
226
229
235
236
245
248
249
250
251


253
255
264
265
266
269
270
275
276
281
285
286
289
290
300
315
321
323
325
327
336
341
345
349
351
360
379
381
382
412
424

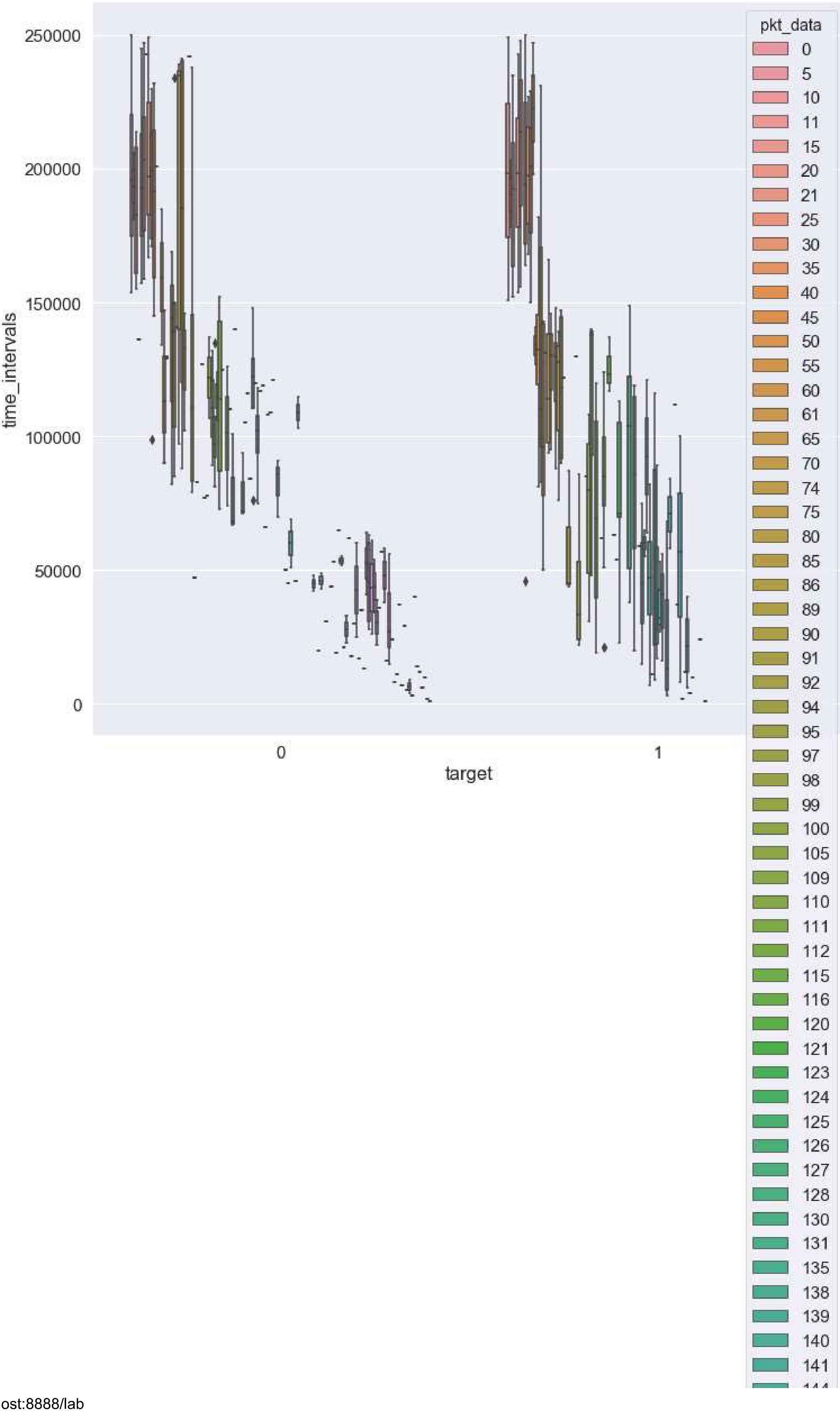
In [40]:

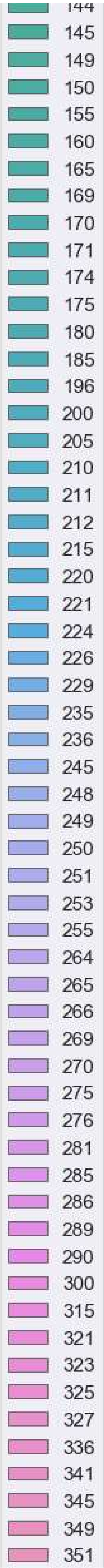
```
plt.figure(figsize=(12,12))
sns.boxplot(x='pkt_data',y='time_intervals',data=dff, hue='target')
plt.show()
```

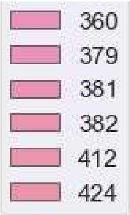


In [41]:

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

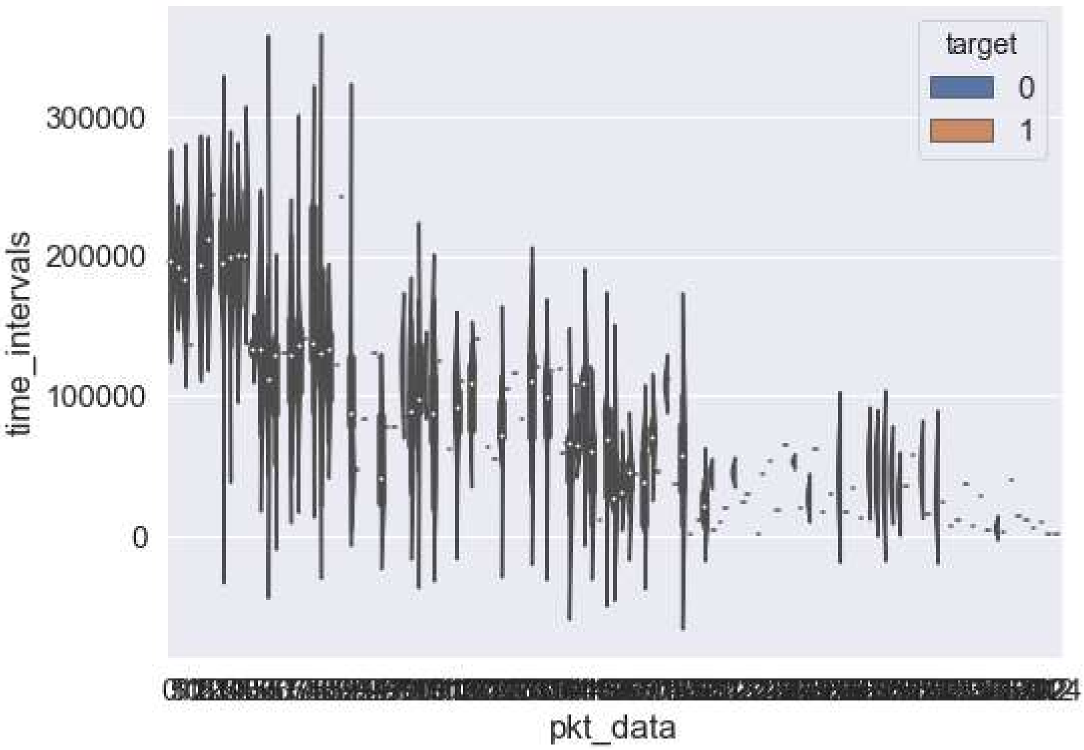

250000
200000
150000
100000
50000
0
time_intervals
0
1
target
pkt_data
0
5
10
11
15
20
21
25
30
35
40
45
50
55
60
61
65
70
74
75
80
85
86
89
90
91
92
94
95
97
98
99
100
105
109
110
111
112
115
116
120
121
123
124
125
126
127
128
130
131
135
138
139
140
141


144
145
149
150
155
160
165
169
170
171
174
175
180
185
196
200
205
210
211
212
215
220
221
224
226
229
235
236
245
248
249
250
251
253
255
264
265
266
269
270
275
276
281
285
286
289
290
300
315
321
323
325
327
336
341
345
349
351



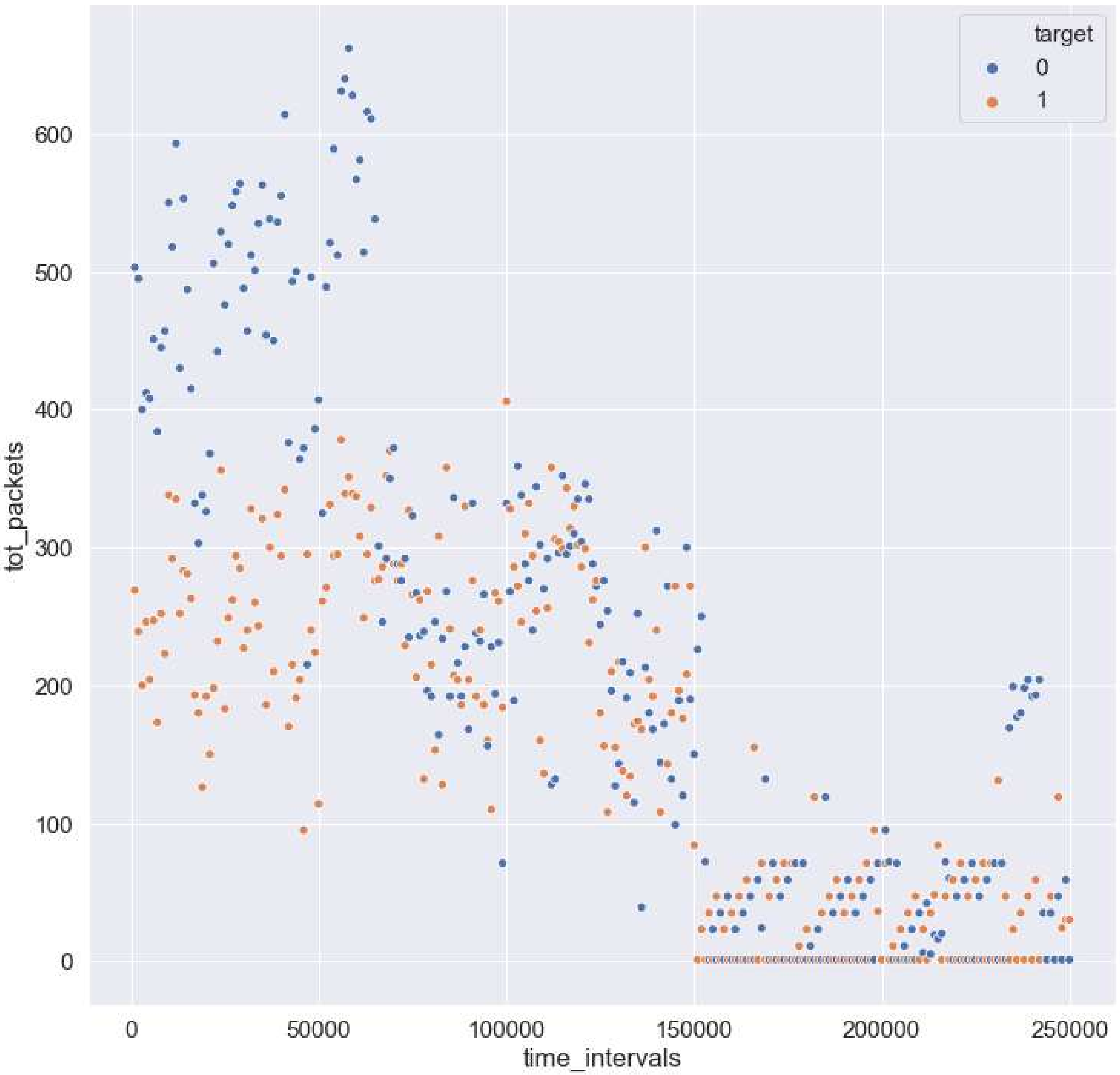
In [42]:

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



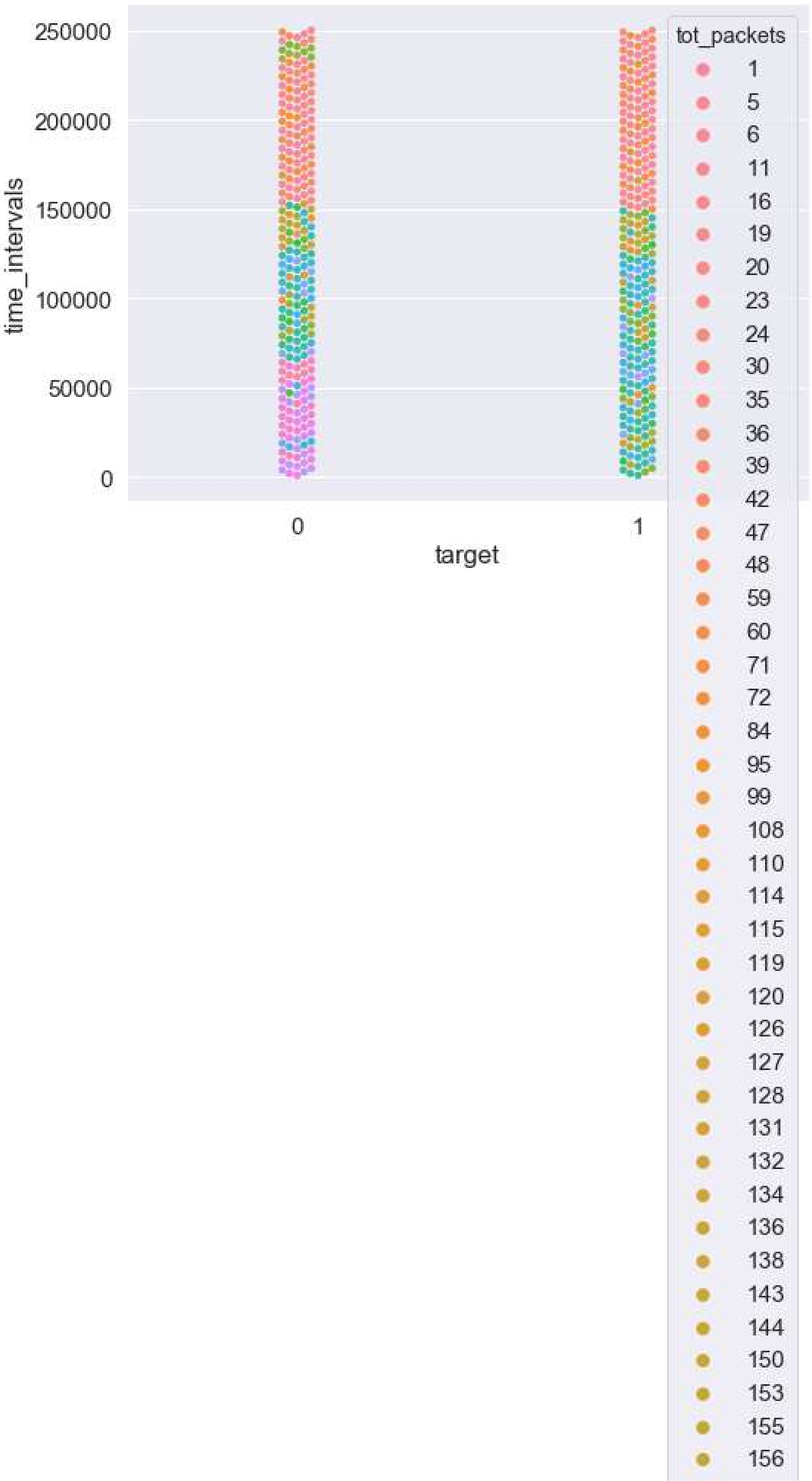
In [43]:

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

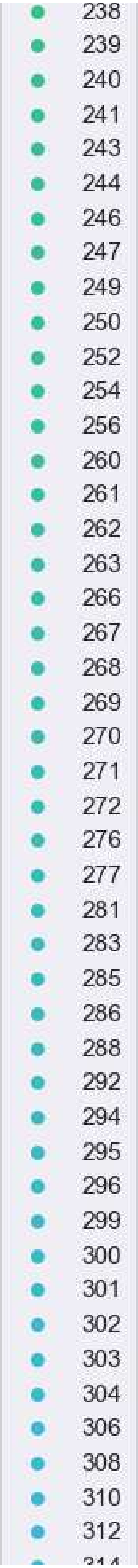


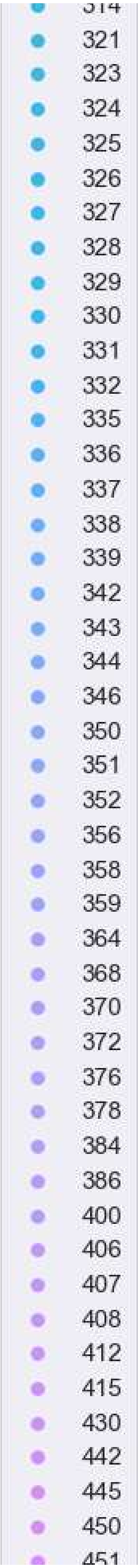
In [44]:

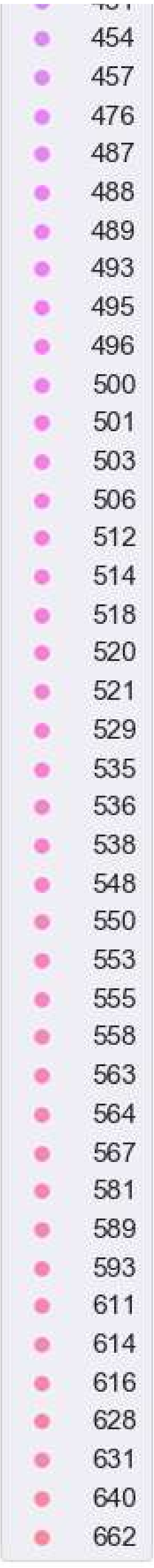
```
plt.figure(figsize=(8,6))
sns.swarmplot(x='target',y='time_intervals',data=dff, hue='tot_packets')
plt.show()
```


250000
200000
150000
100000
50000
0
time_intervals
0
1
target
tot_packets
1
5
6
11
16
19
20
23
24
30
35
36
39
42
47
48
59
60
71
72
84
95
99
108
110
114
115
119
120
126
127
128
131
132
134
136
138
143
144
150
153
155
156

160
164
168
169
170
172
173
174
176
177
180
183
184
186
189
190
191
192
193
194
196
198
199
200
204
206
207
208
209
210
213
215
216
217
223
224
226
227
228
229
231
232
234
235
236

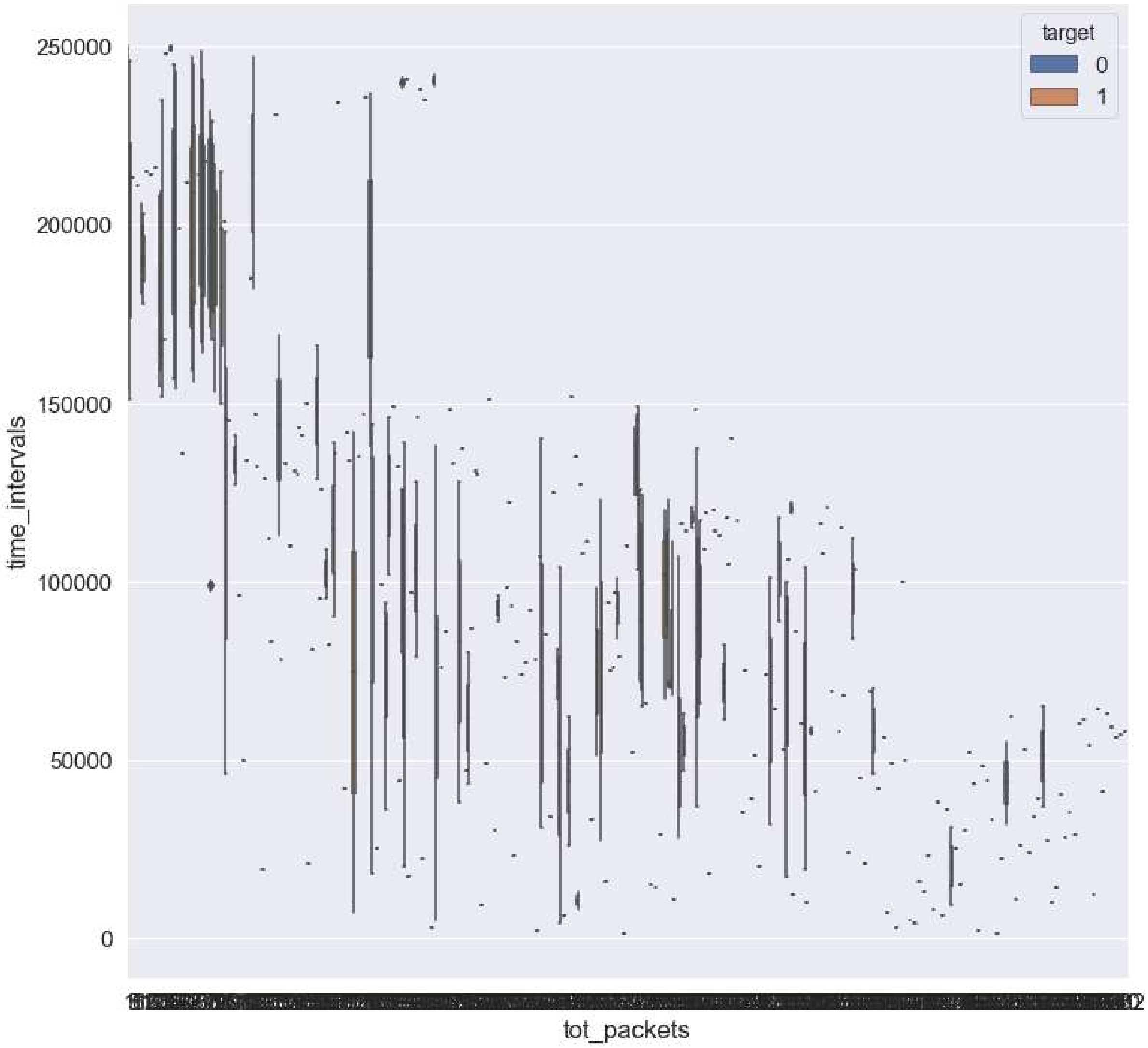

238
239
240
241
243
244
246
247
249
250
252
254
256
260
261
262
263
266
267
268
269
270
271
272
276
277
281
283
285
286
288
292
294
295
296
299
300
301
302
303
304
306
308
310
312
314


314
321
323
324
325
326
327
328
329
330
331
332
335
336
337
338
339
342
343
344
346
350
351
352
356
358
359
364
368
370
372
376
378
384
386
400
406
407
408
412
415
430
442
445
450
451


● 454
● 457
● 476
● 487
● 488
● 489
● 493
● 495
● 496
● 500
● 501
● 503
● 506
● 512
● 514
● 518
● 520
● 521
● 529
● 535
● 536
● 538
● 548
● 550
● 553
● 555
● 558
● 563
● 564
● 567
● 581
● 589
● 593
● 611
● 614
● 616
● 628
● 631
● 640
● 662

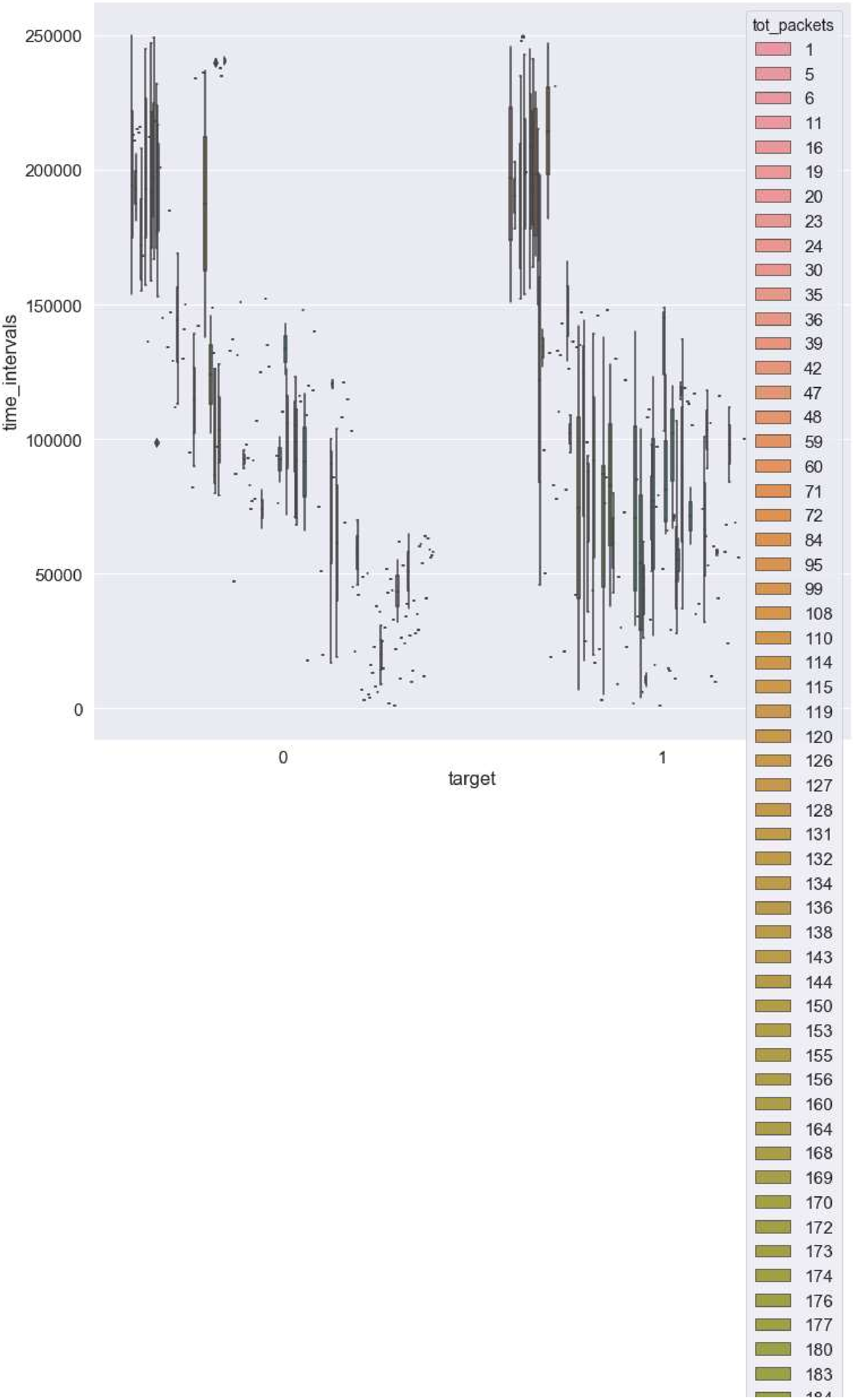
In [45]:

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

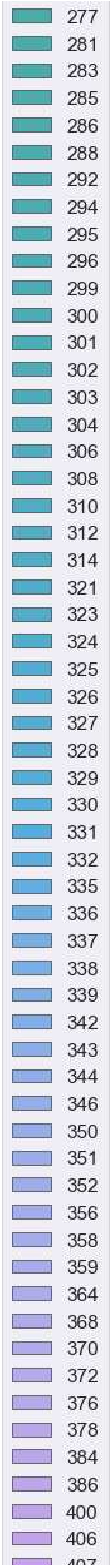


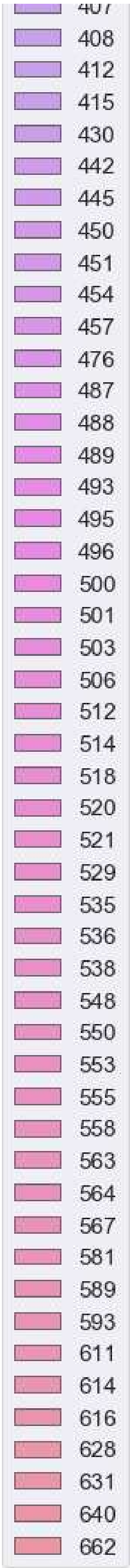
In [46]:

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


tot_packets
1
5
6
11
16
19
20
23
24
30
35
36
39
42
47
48
59
60
71
72
84
95
99
108
110
114
115
119
120
126
127
128
131
132
134
136
138
143
144
150
153
155
156
160
164
168
169
170
172
173
174
176
177
180
183
184
250000
200000
150000
100000
50000
0
time_intervals
0
1
target

184
186
189
190
191
192
193
194
196
198
199
200
204
206
207
208
209
210
213
215
216
217
223
224
226
227
228
229
231
232
234
235
236
238
239
240
241
243
244
246
247
249
250
252
254
256
260
261
262
263
266
267
268
269
270
271
272
276


277
281
283
285
286
288
292
294
295
296
299
300
301
302
303
304
306
308
310
312
314
321
323
324
325
326
327
328
329
330
331
332
335
336
337
338
339
342
343
344
346
350
351
352
356
358
359
364
368
370
372
376
378
384
386
400
406
407


407
408
412
415
430
442
445
450
451
454
457
476
487
488
489
493
495
496
500
501
503
506
512
514
518
520
521
529
535
536
538
548
550
553
555
558
563
564
567
581
589
593
611
614
616
628
631
640
662