EdgeRed Case Study

In [385…
```
# Import Libraries

import pandas as pd
import numpy as np
from collections import Counter
from ydata_profiling import ProfileReport
from datetime import datetime
import matplotlib.pyplot as plt
import seaborn as sb
```

In [3]:
```
# Reading Files

clients = pd.read_csv('C:/Users/anity/Desktop/ER Case Study/Clients.csv')
payments = pd.read_csv('C:/Users/anity/Desktop/ER Case Study/Payments.csv')
```

In [5]:
```
# Reading file clients

clients.head(5)
```

Out[5]:

	client_id	entity_type	entity_year_established
0	786	Australian Private Company	2002
1	230	Australian Private Company	2008
2	282	Individual/Sole Trader	2001
3	447	Australian Private Company	2013
4	310	Individual/Sole Trader	2015

In [7]:
```
# Reading file clients

payments.head(5)
```

Out[7]:

	transaction_id	contract_id	client_id	transaction_date	payment_amt	payment_code
0	20175	927	1	1527012511	66.66	PAYMENT
1	8485	927	1	1511716095	66.66	PAYMENT
2	13778	927	1	1519319303	66.66	PAYMENT
3	22768	927	1	1529863724	66.66	PAYMENT
4	15698	927	1	1521738504	66.66	PAYMENT

In [65]:
```
# Clients file descriptive analysis
print(clients.describe())
print('========>>>>====================>>>>==================>>>>')
print(clients.nunique())
print('========>>>>====================>>>>==================>>>>')
print(clients.entity_type.unique())
```

```
         client_id  entity_year_established
count  1287.000000              1287.000000
mean    641.025641              2009.072261
std     369.778060                 5.708598
min       1.000000              1999.000000
25%     321.500000              2004.000000
50%     640.000000              2010.000000
75%     960.500000              2014.000000
max    1281.000000              2018.000000
========>>>>====================>>>>==================>>>>
client_id                  1281
entity_type                  10
entity_year_established      20
dtype: int64
========>>>>====================>>>>==================>>>>
['Australian Private Company' 'Individual/Sole Trader'
 'Family Partnership' 'Australian Proprietary Company'
 'Discretionary Trading Trust' 'Discretionary Investment Trust'
 'Australian Public Company' 'Other Partnership' 'Fixed Unit Trust'
 'Hybrid Trust']
```

In [66]:
```
# Payments file descriptive analysis
print(payments.describe())
print('========>>>>====================>>>>==================>>>>')
print(payments.nunique())
print('========>>>>====================>>>>==================>>>>')
#print(clients.entity_type.unique())
```

```
       transaction_id   contract_id     client_id  transaction_date  \
count    25559.000000  25559.000000  25559.000000      2.555900e+04
mean     12780.000000    758.221409    602.886811      1.517126e+09
std       7378.392101    352.190207    344.782295      9.964331e+06
min          1.000000      1.000000      1.000000      1.499019e+09
25%       6390.500000    510.000000    308.000000      1.508433e+09
50%      12780.000000    732.000000    593.000000      1.518110e+09
75%      19169.500000    991.000000    894.000000      1.525976e+09
max      25559.000000   1643.000000   1281.000000      1.532456e+09

         payment_amt
count   25559.000000
mean     1221.455691
std      4346.049363
min      -136.660000
25%        93.330000
50%       266.660000
75%       833.330000
max    200000.050000
========>>>>====================>>>>==================>>>>
transaction_id      25559
contract_id          1643
client_id            1281
transaction_date      931
payment_amt           906
payment_code            2
dtype: int64
========>>>>====================>>>>==================>>>>
```

In [11]:
```
# Checking for Anomalies in Clients File using Collections library in clients['c
```

```
Counter(clients['client_id'])
# Shows Multiple entries for Five Clients and rest of them are unique
```

```
Out[11]: Counter({591: 3,
                  473: 2,
                  165: 2,
                  797: 2,
                  1262: 2,
                  786: 1,
                  230: 1,
                  282: 1,
                  447: 1,
                  310: 1,
                  539: 1,
                  744: 1,
                  920: 1,
                  1104: 1,
                  1187: 1,
                  534: 1,
                  810: 1,
                  678: 1,
                  834: 1,
                  551: 1,
                  97: 1,
                  630: 1,
                  1042: 1,
                  743: 1,
                  55: 1,
                  535: 1,
                  1179: 1,
                  242: 1,
                  311: 1,
                  725: 1,
                  510: 1,
                  876: 1,
                  1272: 1,
                  216: 1,
                  43: 1,
                  571: 1,
                  953: 1,
                  950: 1,
                  452: 1,
                  1277: 1,
                  831: 1,
                  706: 1,
                  909: 1,
                  829: 1,
                  241: 1,
                  1143: 1,
                  469: 1,
                  259: 1,
                  410: 1,
                  514: 1,
                  871: 1,
                  1094: 1,
                  256: 1,
                  975: 1,
                  414: 1,
                  966: 1,
                  941: 1,
                  1114: 1,
                  313: 1,
                  1214: 1,
```

```
359: 1,
498: 1,
58: 1,
726: 1,
987: 1,
646: 1,
677: 1,
222: 1,
411: 1,
588: 1,
172: 1,
200: 1,
531: 1,
600: 1,
937: 1,
910: 1,
919: 1,
541: 1,
405: 1,
991: 1,
758: 1,
274: 1,
754: 1,
528: 1,
873: 1,
373: 1,
465: 1,
520: 1,
279: 1,
390: 1,
456: 1,
875: 1,
436: 1,
576: 1,
24: 1,
2: 1,
374: 1,
1267: 1,
49: 1,
1038: 1,
1227: 1,
11: 1,
396: 1,
379: 1,
344: 1,
818: 1,
1035: 1,
954: 1,
1224: 1,
275: 1,
665: 1,
503: 1,
238: 1,
1062: 1,
65: 1,
317: 1,
878: 1,
1017: 1,
544: 1,
163: 1,
```

92: 1,
990: 1,
179: 1,
108: 1,
961: 1,
759: 1,
687: 1,
1101: 1,
732: 1,
370: 1,
1121: 1,
1184: 1,
124: 1,
120: 1,
700: 1,
888: 1,
1019: 1,
670: 1,
525: 1,
983: 1,
383: 1,
587: 1,
497: 1,
927: 1,
827: 1,
982: 1,
1124: 1,
593: 1,
807: 1,
853: 1,
1231: 1,
1273: 1,
581: 1,
199: 1,
451: 1,
496: 1,
679: 1,
225: 1,
401: 1,
832: 1,
437: 1,
565: 1,
518: 1,
1058: 1,
291: 1,
574: 1,
1103: 1,
104: 1,
949: 1,
103: 1,
816: 1,
1000: 1,
365: 1,
1028: 1,
139: 1,
713: 1,
283: 1,
271: 1,
223: 1,
985: 1,

```
1119: 1,
505: 1,
566: 1,
651: 1,
1243: 1,
908: 1,
537: 1,
780: 1,
839: 1,
218: 1,
478: 1,
176: 1,
453: 1,
674: 1,
841: 1,
144: 1,
771: 1,
119: 1,
1033: 1,
219: 1,
418: 1,
1053: 1,
648: 1,
615: 1,
861: 1,
594: 1,
938: 1,
925: 1,
429: 1,
1176: 1,
766: 1,
1086: 1,
215: 1,
507: 1,
773: 1,
711: 1,
907: 1,
890: 1,
1003: 1,
874: 1,
23: 1,
886: 1,
730: 1,
784: 1,
911: 1,
867: 1,
324: 1,
800: 1,
308: 1,
792: 1,
1027: 1,
1166: 1,
134: 1,
466: 1,
851: 1,
481: 1,
879: 1,
833: 1,
1263: 1,
1264: 1,
```

695: 1,
798: 1,
753: 1,
1132: 1,
81: 1,
812: 1,
364: 1,
1134: 1,
188: 1,
375: 1,
1014: 1,
1125: 1,
9: 1,
796: 1,
1006: 1,
150: 1,
336: 1,
233: 1,
1258: 1,
578: 1,
40: 1,
682: 1,
989: 1,
617: 1,
626: 1,
39: 1,
121: 1,
970: 1,
502: 1,
859: 1,
824: 1,
1100: 1,
1022: 1,
304: 1,
15: 1,
659: 1,
584: 1,
666: 1,
499: 1,
143: 1,
455: 1,
660: 1,
1196: 1,
972: 1,
1168: 1,
536: 1,
644: 1,
791: 1,
126: 1,
524: 1,
764: 1,
53: 1,
1117: 1,
210: 1,
1129: 1,
788: 1,
1141: 1,
177: 1,
618: 1,
166: 1,

728: 1,
84: 1,
250: 1,
229: 1,
480: 1,
870: 1,
378: 1,
573: 1,
468: 1,
328: 1,
1153: 1,
288: 1,
354: 1,
951: 1,
118: 1,
1161: 1,
515: 1,
597: 1,
500: 1,
1088: 1,
73: 1,
772: 1,
633: 1,
388: 1,
720: 1,
1041: 1,
734: 1,
689: 1,
893: 1,
1219: 1,
397: 1,
392: 1,
338: 1,
1160: 1,
636: 1,
351: 1,
187: 1,
569: 1,
440: 1,
662: 1,
790: 1,
563: 1,
195: 1,
261: 1,
968: 1,
4: 1,
1280: 1,
389: 1,
1223: 1,
394: 1,
446: 1,
1256: 1,
710: 1,
624: 1,
683: 1,
371: 1,
663: 1,
251: 1,
1067: 1,
554: 1,

```
1152: 1,
702: 1,
809: 1,
641: 1,
318: 1,
286: 1,
1049: 1,
70: 1,
750: 1,
789: 1,
1147: 1,
799: 1,
842: 1,
1020: 1,
556: 1,
1178: 1,
540: 1,
860: 1,
1175: 1,
109: 1,
1274: 1,
579: 1,
762: 1,
320: 1,
450: 1,
782: 1,
988: 1,
669: 1,
428: 1,
347: 1,
708: 1,
182: 1,
738: 1,
709: 1,
862: 1,
977: 1,
794: 1,
13: 1,
204: 1,
849: 1,
62: 1,
1172: 1,
1271: 1,
349: 1,
650: 1,
545: 1,
424: 1,
1169: 1,
1170: 1,
529: 1,
628: 1,
1078: 1,
1097: 1,
1081: 1,
745: 1,
1113: 1,
777: 1,
1032: 1,
439: 1,
99: 1,
```

1057: 1,
211: 1,
33: 1,
94: 1,
793: 1,
152: 1,
1005: 1,
181: 1,
1118: 1,
122: 1,
281: 1,
608: 1,
296: 1,
185: 1,
433: 1,
180: 1,
1151: 1,
1140: 1,
125: 1,
572: 1,
1149: 1,
333: 1,
47: 1,
684: 1,
819: 1,
1211: 1,
45: 1,
140: 1,
184: 1,
1246: 1,
1085: 1,
474: 1,
153: 1,
693: 1,
6: 1,
80: 1,
323: 1,
705: 1,
930: 1,
856: 1,
156: 1,
1107: 1,
1164: 1,
1106: 1,
1076: 1,
205: 1,
145: 1,
1025: 1,
622: 1,
668: 1,
327: 1,
1031: 1,
501: 1,
422: 1,
1045: 1,
1023: 1,
341: 1,
517: 1,
1064: 1,
1096: 1,

```
1183: 1,
742: 1,
417: 1,
415: 1,
1201: 1,
434: 1,
707: 1,
253: 1,
1089: 1,
284: 1,
467: 1,
882: 1,
421: 1,
1180: 1,
959: 1,
1270: 1,
986: 1,
1210: 1,
855: 1,
1217: 1,
244: 1,
1116: 1,
420: 1,
830: 1,
616: 1,
942: 1,
398: 1,
112: 1,
68: 1,
132: 1,
844: 1,
1135: 1,
37: 1,
246: 1,
1083: 1,
1216: 1,
1173: 1,
331: 1,
1279: 1,
1222: 1,
822: 1,
838: 1,
1142: 1,
601: 1,
609: 1,
549: 1,
1158: 1,
159: 1,
637: 1,
1202: 1,
564: 1,
892: 1,
509: 1,
653: 1,
1191: 1,
63: 1,
192: 1,
1055: 1,
358: 1,
479: 1,
```

```
538: 1,
334: 1,
1165: 1,
482: 1,
521: 1,
1021: 1,
1254: 1,
887: 1,
1265: 1,
1056: 1,
25: 1,
385: 1,
586: 1,
939: 1,
76: 1,
699: 1,
889: 1,
1066: 1,
1099: 1,
161: 1,
287: 1,
661: 1,
314: 1,
1159: 1,
1197: 1,
350: 1,
778: 1,
1221: 1,
337: 1,
130: 1,
896: 1,
1052: 1,
203: 1,
899: 1,
1108: 1,
1047: 1,
1011: 1,
460: 1,
462: 1,
657: 1,
298: 1,
1190: 1,
947: 1,
787: 1,
715: 1,
1004: 1,
21: 1,
553: 1,
826: 1,
926: 1,
1043: 1,
913: 1,
1189: 1,
78: 1,
164: 1,
962: 1,
735: 1,
1206: 1,
1093: 1,
655: 1,
```

```
1193: 1,
268: 1,
884: 1,
751: 1,
1092: 1,
513: 1,
776: 1,
306: 1,
530: 1,
61: 1,
169: 1,
267: 1,
932: 1,
767: 1,
302: 1,
491: 1,
1080: 1,
739: 1,
1069: 1,
580: 1,
483: 1,
731: 1,
1157: 1,
431: 1,
20: 1,
639: 1,
129: 1,
964: 1,
1203: 1,
625: 1,
846: 1,
366: 1,
191: 1,
635: 1,
774: 1,
843: 1,
128: 1,
160: 1,
1001: 1,
582: 1,
685: 1,
675: 1,
329: 1,
272: 1,
557: 1,
673: 1,
217: 1,
1199: 1,
423: 1,
441: 1,
117: 1,
872: 1,
645: 1,
303: 1,
293: 1,
88: 1,
652: 1,
444: 1,
781: 1,
1225: 1,
```

```
402: 1,
733: 1,
170: 1,
642: 1,
1145: 1,
1248: 1,
1150: 1,
1120: 1,
795: 1,
340: 1,
362: 1,
696: 1,
1240: 1,
312: 1,
339: 1,
740: 1,
924: 1,
1171: 1,
301: 1,
611: 1,
940: 1,
1198: 1,
596: 1,
470: 1,
1138: 1,
361: 1,
877: 1,
237: 1,
676: 1,
716: 1,
1268: 1,
868: 1,
1177: 1,
110: 1,
1205: 1,
1215: 1,
264: 1,
946: 1,
168: 1,
485: 1,
632: 1,
560: 1,
575: 1,
355: 1,
262: 1,
1029: 1,
1226: 1,
155: 1,
996: 1,
820: 1,
1148: 1,
1082: 1,
577: 1,
213: 1,
162: 1,
427: 1,
901: 1,
29: 1,
77: 1,
137: 1,
```

638: 1,
967: 1,
1060: 1,
299: 1,
82: 1,
836: 1,
91: 1,
95: 1,
1232: 1,
158: 1,
147: 1,
604: 1,
943: 1,
1128: 1,
649: 1,
309: 1,
769: 1,
558: 1,
345: 1,
903: 1,
101: 1,
377: 1,
1050: 1,
363: 1,
749: 1,
1230: 1,
814: 1,
923: 1,
729: 1,
1102: 1,
547: 1,
952: 1,
1229: 1,
438: 1,
154: 1,
756: 1,
1181: 1,
1070: 1,
278: 1,
717: 1,
369: 1,
1212: 1,
1071: 1,
348: 1,
56: 1,
550: 1,
1016: 1,
722: 1,
703: 1,
197: 1,
697: 1,
599: 1,
1068: 1,
266: 1,
727: 1,
865: 1,
1077: 1,
532: 1,
48: 1,
747: 1,

667: 1,
1075: 1,
821: 1,
346: 1,
127: 1,
459: 1,
813: 1,
59: 1,
409: 1,
543: 1,
1244: 1,
1235: 1,
464: 1,
231: 1,
1281: 1,
555: 1,
270: 1,
770: 1,
1084: 1,
610: 1,
38: 1,
400: 1,
1156: 1,
475: 1,
1194: 1,
96: 1,
854: 1,
387: 1,
973: 1,
367: 1,
202: 1,
1015: 1,
391: 1,
407: 1,
1: 1,
519: 1,
22: 1,
1259: 1,
690: 1,
1242: 1,
102: 1,
1098: 1,
1204: 1,
656: 1,
357: 1,
922: 1,
1237: 1,
1278: 1,
131: 1,
227: 1,
380: 1,
1044: 1,
280: 1,
629: 1,
718: 1,
30: 1,
585: 1,
41: 1,
42: 1,
746: 1,

297: 1,
894: 1,
900: 1,
135: 1,
123: 1,
523: 1,
589: 1,
19: 1,
319: 1,
1012: 1,
704: 1,
189: 1,
858: 1,
590: 1,
995: 1,
701: 1,
232: 1,
245: 1,
692: 1,
1234: 1,
484: 1,
546: 1,
235: 1,
607: 1,
35: 1,
721: 1,
891: 1,
248: 1,
243: 1,
67: 1,
880: 1,
897: 1,
1115: 1,
100: 1,
603: 1,
448: 1,
935: 1,
289: 1,
848: 1,
881: 1,
435: 1,
906: 1,
368: 1,
912: 1,
1252: 1,
46: 1,
416: 1,
934: 1,
443: 1,
527: 1,
559: 1,
654: 1,
8: 1,
133: 1,
1260: 1,
1136: 1,
477: 1,
631: 1,
1036: 1,
806: 1,

463: 1,
107: 1,
157: 1,
249: 1,
1182: 1,
775: 1,
1091: 1,
997: 1,
944: 1,
723: 1,
857: 1,
226: 1,
258: 1,
1247: 1,
472: 1,
1051: 1,
183: 1,
724: 1,
381: 1,
980: 1,
984: 1,
1154: 1,
85: 1,
1208: 1,
1024: 1,
845: 1,
32: 1,
811: 1,
175: 1,
316: 1,
1276: 1,
79: 1,
1122: 1,
915: 1,
613: 1,
567: 1,
614: 1,
840: 1,
955: 1,
1137: 1,
905: 1,
931: 1,
1126: 1,
570: 1,
488: 1,
353: 1,
904: 1,
395: 1,
300: 1,
1186: 1,
1061: 1,
1026: 1,
623: 1,
994: 1,
71: 1,
393: 1,
1238: 1,
220: 1,
360: 1,
533: 1,

```
 322: 1,
 1163: 1,
 263: 1,
 208: 1,
 236: 1,
 1241: 1,
 1195: 1,
 412: 1,
 1155: 1,
 965: 1,
 1133: 1,
 90: 1,
 254: 1,
 1139: 1,
 193: 1,
 562: 1,
 51: 1,
 489: 1,
 658: 1,
 1123: 1,
 60: 1,
 50: 1,
 461: 1,
 1261: 1,
 1074: 1,
 785: 1,
 207: 1,
 835: 1,
 1245: 1,
 1063: 1,
 801: 1,
 342: 1,
 671: 1,
 760: 1,
 976: 1,
 748: 1,
 495: 1,
 44: 1,
 1209: 1,
 86: 1,
 ...})
```

In [38]:
```
# Overhecking again for Anomalies in Clients File using groupby function in clie

clients.groupby('client_id')['client_id'].count().sort_values()
```

Out[38]:
```
client_id
1       1
858     1
857     1
856     1
855     1
       ..
473     2
1262    2
165     2
797     2
591     3
Name: client_id, Length: 1281, dtype: int64
```

In [67]:
```
# Checking for Anomalies in Payments File using Collections library in clients['

Counter(payments['client_id'])
# Shows Several uneven entries for all the values
```

Out[67]:
```
Counter({413: 105,
         859: 92,
         1124: 81,
         777: 79,
         591: 78,
         1128: 78,
         726: 77,
         231: 76,
         569: 75,
         718: 75,
         795: 72,
         289: 70,
         283: 68,
         995: 68,
         463: 67,
         1126: 67,
         571: 66,
         93: 65,
         797: 65,
         916: 65,
         1007: 65,
         705: 62,
         969: 62,
         903: 61,
         1080: 61,
         695: 59,
         775: 58,
         1053: 58,
         8: 56,
         190: 56,
         721: 56,
         753: 56,
         208: 55,
         522: 55,
         177: 54,
         786: 54,
         1094: 53,
         25: 52,
         257: 52,
         331: 52,
         792: 52,
         871: 52,
         355: 51,
         808: 51,
         209: 50,
         685: 50,
         958: 50,
         367: 49,
         46: 48,
         56: 48,
         258: 48,
         338: 48,
         678: 48,
         371: 47,
         470: 46,
         523: 46,
         937: 46,
         953: 46,
         316: 45,
         344: 45,
```

```
30: 44,
168: 44,
205: 44,
244: 44,
254: 44,
713: 44,
1071: 44,
130: 43,
388: 43,
634: 43,
770: 43,
110: 42,
159: 42,
237: 42,
280: 42,
328: 42,
534: 42,
708: 42,
902: 42,
55: 41,
199: 41,
279: 41,
606: 41,
1033: 41,
1135: 41,
1195: 41,
366: 40,
703: 40,
719: 40,
799: 40,
919: 40,
1103: 40,
78: 39,
538: 39,
589: 39,
105: 38,
278: 38,
281: 38,
327: 38,
332: 38,
365: 38,
434: 38,
525: 38,
537: 38,
542: 38,
557: 38,
747: 38,
853: 38,
890: 38,
891: 38,
1042: 38,
1025: 37,
1098: 37,
101: 36,
132: 36,
224: 36,
399: 36,
423: 36,
458: 36,
555: 36,
```

```
621: 36,
690: 36,
711: 36,
723: 36,
804: 36,
864: 36,
908: 36,
914: 36,
921: 36,
964: 36,
979: 36,
983: 36,
79: 34,
99: 34,
145: 34,
165: 34,
166: 34,
195: 34,
226: 34,
285: 34,
286: 34,
297: 34,
308: 34,
313: 34,
398: 34,
459: 34,
511: 34,
524: 34,
536: 34,
592: 34,
623: 34,
649: 34,
654: 34,
763: 34,
767: 34,
843: 34,
885: 34,
1024: 34,
216: 33,
469: 33,
531: 33,
1076: 33,
53: 32,
92: 32,
146: 32,
167: 32,
182: 32,
277: 32,
290: 32,
292: 32,
385: 32,
411: 32,
417: 32,
445: 32,
533: 32,
540: 32,
582: 32,
724: 32,
745: 32,
778: 32,
```

```
789: 32,
791: 32,
798: 32,
810: 32,
858: 32,
872: 32,
884: 32,
915: 32,
927: 32,
947: 32,
991: 32,
1038: 32,
1058: 32,
1081: 32,
1121: 32,
1198: 32,
248: 31,
336: 31,
407: 31,
415: 31,
849: 31,
27: 30,
38: 30,
41: 30,
54: 30,
57: 30,
72: 30,
88: 30,
115: 30,
129: 30,
143: 30,
149: 30,
163: 30,
239: 30,
241: 30,
260: 30,
282: 30,
291: 30,
301: 30,
309: 30,
322: 30,
324: 30,
337: 30,
359: 30,
361: 30,
368: 30,
383: 30,
409: 30,
427: 30,
431: 30,
455: 30,
457: 30,
467: 30,
472: 30,
473: 30,
484: 30,
487: 30,
489: 30,
501: 30,
516: 30,
```

```
521: 30,
527: 30,
546: 30,
547: 30,
562: 30,
574: 30,
598: 30,
605: 30,
608: 30,
624: 30,
628: 30,
646: 30,
650: 30,
658: 30,
679: 30,
697: 30,
707: 30,
731: 30,
742: 30,
761: 30,
782: 30,
790: 30,
812: 30,
855: 30,
861: 30,
883: 30,
889: 30,
898: 30,
909: 30,
911: 30,
967: 30,
1004: 30,
1036: 30,
1046: 30,
1069: 30,
1087: 30,
1093: 30,
1105: 30,
1115: 30,
1125: 30,
1130: 30,
7: 29,
369: 29,
617: 29,
667: 29,
766: 29,
874: 29,
11: 28,
33: 28,
39: 28,
43: 28,
65: 28,
76: 28,
94: 28,
102: 28,
109: 28,
113: 28,
144: 28,
186: 28,
189: 28,
```

```
193: 28,
194: 28,
215: 28,
227: 28,
293: 28,
294: 28,
299: 28,
325: 28,
339: 28,
347: 28,
381: 28,
389: 28,
429: 28,
451: 28,
464: 28,
495: 28,
509: 28,
514: 28,
552: 28,
554: 28,
566: 28,
585: 28,
586: 28,
622: 28,
642: 28,
648: 28,
661: 28,
665: 28,
674: 28,
682: 28,
683: 28,
686: 28,
688: 28,
701: 28,
704: 28,
714: 28,
715: 28,
722: 28,
727: 28,
735: 28,
744: 28,
787: 28,
801: 28,
809: 28,
840: 28,
866: 28,
880: 28,
938: 28,
942: 28,
951: 28,
959: 28,
966: 28,
985: 28,
989: 28,
994: 28,
1000: 28,
1060: 28,
1072: 28,
1089: 28,
1108: 28,
```

```
1110: 28,
1114: 28,
1122: 28,
1134: 28,
112: 27,
342: 27,
544: 27,
781: 27,
962: 27,
17: 26,
29: 26,
40: 26,
42: 26,
74: 26,
91: 26,
98: 26,
127: 26,
131: 26,
133: 26,
176: 26,
188: 26,
219: 26,
228: 26,
229: 26,
250: 26,
255: 26,
270: 26,
275: 26,
284: 26,
303: 26,
326: 26,
334: 26,
349: 26,
350: 26,
353: 26,
356: 26,
412: 26,
414: 26,
416: 26,
428: 26,
430: 26,
444: 26,
446: 26,
460: 26,
461: 26,
517: 26,
518: 26,
519: 26,
543: 26,
563: 26,
573: 26,
604: 26,
666: 26,
671: 26,
706: 26,
734: 26,
773: 26,
794: 26,
807: 26,
841: 26,
```

```
867: 26,
878: 26,
887: 26,
894: 26,
899: 26,
920: 26,
939: 26,
948: 26,
997: 26,
1023: 26,
1028: 26,
1043: 26,
1052: 26,
1055: 26,
1073: 26,
1084: 26,
1088: 26,
1092: 26,
1104: 26,
1116: 26,
1120: 26,
468: 25,
490: 25,
545: 25,
857: 25,
1015: 25,
1021: 25,
10: 24,
18: 24,
70: 24,
100: 24,
103: 24,
120: 24,
136: 24,
142: 24,
156: 24,
170: 24,
214: 24,
218: 24,
263: 24,
268: 24,
317: 24,
318: 24,
345: 24,
360: 24,
376: 24,
380: 24,
390: 24,
449: 24,
452: 24,
453: 24,
479: 24,
499: 24,
515: 24,
520: 24,
526: 24,
556: 24,
567: 24,
581: 24,
583: 24,
```

```
584: 24,
638: 24,
643: 24,
670: 24,
738: 24,
748: 24,
750: 24,
759: 24,
830: 24,
834: 24,
837: 24,
863: 24,
888: 24,
910: 24,
923: 24,
929: 24,
932: 24,
975: 24,
982: 24,
1018: 24,
1039: 24,
1047: 24,
1057: 24,
1070: 24,
1075: 24,
1193: 24,
13: 23,
233: 23,
785: 23,
806: 23,
4: 22,
6: 22,
32: 22,
45: 22,
63: 22,
67: 22,
81: 22,
87: 22,
97: 22,
139: 22,
140: 22,
161: 22,
171: 22,
221: 22,
245: 22,
247: 22,
304: 22,
311: 22,
378: 22,
382: 22,
386: 22,
393: 22,
396: 22,
402: 22,
408: 22,
418: 22,
432: 22,
438: 22,
505: 22,
559: 22,
```

```
577: 22,
578: 22,
580: 22,
593: 22,
653: 22,
664: 22,
668: 22,
772: 22,
800: 22,
802: 22,
829: 22,
838: 22,
856: 22,
892: 22,
904: 22,
913: 22,
917: 22,
946: 22,
949: 22,
960: 22,
974: 22,
1027: 22,
1029: 22,
1063: 22,
1068: 22,
1123: 22,
1131: 22,
83: 21,
259: 21,
320: 21,
539: 21,
3: 20,
22: 20,
47: 20,
51: 20,
64: 20,
77: 20,
89: 20,
96: 20,
116: 20,
117: 20,
119: 20,
155: 20,
251: 20,
265: 20,
276: 20,
287: 20,
305: 20,
306: 20,
315: 20,
340: 20,
363: 20,
397: 20,
448: 20,
475: 20,
476: 20,
494: 20,
502: 20,
503: 20,
507: 20,
```

```
528: 20,
549: 20,
587: 20,
595: 20,
631: 20,
656: 20,
676: 20,
681: 20,
691: 20,
693: 20,
699: 20,
776: 20,
796: 20,
817: 20,
827: 20,
895: 20,
925: 20,
934: 20,
970: 20,
1010: 20,
1040: 20,
1062: 20,
1077: 20,
1083: 20,
1085: 20,
1136: 20,
1139: 20,
211: 19,
876: 19,
945: 19,
1: 18,
24: 18,
35: 18,
61: 18,
223: 18,
225: 18,
234: 18,
236: 18,
243: 18,
262: 18,
274: 18,
333: 18,
352: 18,
377: 18,
508: 18,
607: 18,
612: 18,
614: 18,
632: 18,
655: 18,
737: 18,
749: 18,
765: 18,
768: 18,
771: 18,
774: 18,
828: 18,
833: 18,
852: 18,
875: 18,
```

```
879: 18,
922: 18,
955: 18,
965: 18,
973: 18,
988: 18,
1002: 18,
1035: 18,
1082: 18,
1096: 18,
1101: 18,
1102: 18,
1178: 18,
1210: 18,
462: 17,
756: 17,
26: 16,
73: 16,
153: 16,
158: 16,
169: 16,
184: 16,
213: 16,
261: 16,
321: 16,
357: 16,
364: 16,
391: 16,
400: 16,
433: 16,
442: 16,
466: 16,
482: 16,
513: 16,
560: 16,
588: 16,
629: 16,
633: 16,
635: 16,
641: 16,
696: 16,
709: 16,
720: 16,
740: 16,
754: 16,
783: 16,
816: 16,
818: 16,
820: 16,
832: 16,
836: 16,
901: 16,
931: 16,
935: 16,
963: 16,
972: 16,
976: 16,
1005: 16,
1006: 16,
1019: 16,
```

```
1026: 16,
1044: 16,
1078: 16,
1079: 16,
1111: 16,
1142: 16,
1162: 16,
1172: 16,
1173: 16,
1222: 16,
736: 15,
14: 14,
19: 14,
49: 14,
66: 14,
75: 14,
82: 14,
118: 14,
138: 14,
147: 14,
148: 14,
160: 14,
174: 14,
180: 14,
222: 14,
323: 14,
343: 14,
370: 14,
374: 14,
425: 14,
435: 14,
474: 14,
477: 14,
504: 14,
561: 14,
570: 14,
611: 14,
639: 14,
680: 14,
689: 14,
716: 14,
717: 14,
764: 14,
839: 14,
854: 14,
869: 14,
933: 14,
956: 14,
968: 14,
993: 14,
1009: 14,
1032: 14,
1037: 14,
1050: 14,
1051: 14,
1109: 14,
1137: 14,
1138: 14,
1140: 14,
1146: 14,
```

```
1213: 14,
373: 13,
16: 12,
20: 12,
21: 12,
62: 12,
80: 12,
90: 12,
123: 12,
150: 12,
162: 12,
172: 12,
173: 12,
198: 12,
202: 12,
256: 12,
341: 12,
346: 12,
372: 12,
379: 12,
392: 12,
419: 12,
426: 12,
439: 12,
492: 12,
496: 12,
506: 12,
512: 12,
548: 12,
601: 12,
619: 12,
651: 12,
669: 12,
684: 12,
692: 12,
732: 12,
751: 12,
814: 12,
831: 12,
848: 12,
850: 12,
907: 12,
940: 12,
941: 12,
944: 12,
992: 12,
999: 12,
1012: 12,
1014: 12,
1016: 12,
1041: 12,
1059: 12,
1074: 12,
1099: 12,
1106: 12,
1113: 12,
1127: 12,
1141: 12,
1143: 12,
1148: 12,
```

```
1157: 12,
1165: 12,
1174: 12,
1185: 12,
1186: 12,
1189: 12,
1190: 12,
1212: 12,
1225: 12,
1241: 12,
1254: 12,
126: 11,
613: 11,
687: 11,
31: 10,
86: 10,
124: 10,
137: 10,
141: 10,
151: 10,
152: 10,
183: 10,
187: 10,
204: 10,
207: 10,
210: 10,
232: 10,
252: 10,
271: 10,
319: 10,
348: 10,
354: 10,
375: 10,
420: 10,
465: 10,
480: 10,
493: 10,
497: 10,
498: 10,
535: 10,
553: 10,
594: 10,
596: 10,
603: 10,
620: 10,
627: 10,
630: 10,
636: 10,
640: 10,
647: 10,
662: 10,
710: 10,
752: 10,
803: 10,
811: 10,
815: 10,
822: 10,
873: 10,
886: 10,
896: 10,
```

```
906: 10,
943: 10,
950: 10,
998: 10,
1034: 10,
1049: 10,
1064: 10,
1100: 10,
1112: 10,
1117: 10,
1129: 10,
1132: 10,
1133: 10,
1145: 10,
1147: 10,
1153: 10,
1171: 10,
1175: 10,
1181: 10,
1188: 10,
1191: 10,
1192: 10,
1196: 10,
1197: 10,
1199: 10,
1201: 10,
1202: 10,
1203: 10,
1204: 10,
1207: 10,
1209: 10,
1211: 10,
1214: 10,
1240: 10,
1245: 10,
1252: 10,
1262: 10,
456: 9,
870: 9,
2: 8,
28: 8,
34: 8,
44: 8,
48: 8,
50: 8,
52: 8,
107: 8,
108: 8,
128: 8,
135: 8,
157: 8,
191: 8,
196: 8,
212: 8,
235: 8,
288: 8,
296: 8,
302: 8,
310: 8,
312: 8,
```

```
                330: 8,
                351: 8,
                394: 8,
                395: 8,
                404: 8,
                424: 8,
                440: 8,
                447: 8,
                481: 8,
                485: 8,
                488: 8,
                491: 8,
                551: 8,
                558: 8,
                564: 8,
                568: 8,
                576: 8,
                579: 8,
                590: 8,
                597: 8,
                599: 8,
                637: 8,
                645: 8,
                663: 8,
                672: 8,
                700: 8,
                702: 8,
                728: 8,
                730: 8,
                821: 8,
                824: 8,
                844: 8,
                845: 8,
                847: 8,
                862: 8,
                912: 8,
                926: 8,
                928: 8,
                930: 8,
                952: 8,
                ...})
```

In [69]:
```
# Overhecking again for Anomalies in payments File using groupby function in cli

payments.groupby('client_id')['client_id'].count().sort_values()
```

Out[69]:
```
client_id
134       1
387       1
1281      2
757       2
762       2
        ...
1128     78
777      79
1124     81
859      92
413     105
Name: client_id, Length: 1281, dtype: int64
```

Now Trying to merge both the files using merge()

In [71]:
```
payments.merge(clients, on='client_id', how='left')
```

Out[71]:

	transaction_id	contract_id	client_id	transaction_date	payment_amt	payment_code	er
0	20175	927	1	1527012511	66.66	PAYMENT	P
1	8485	927	1	1511716095	66.66	PAYMENT	P
2	13778	927	1	1519319303	66.66	PAYMENT	P
3	22768	927	1	1529863724	66.66	PAYMENT	P
4	15698	927	1	1521738504	66.66	PAYMENT	P
...	...	...	...	...	...	...	
25849	25075	1603	1280	1532023764	1666.68	PAYMENT	
25850	24711	1603	1280	1531764560	0.01	PAYMENT	
25851	25076	1603	1280	1532023764	64.99	PAYMENT	
25852	25132	1627	1281	1532282886	0.01	PAYMENT	
25853	25131	1627	1281	1532282886	20000.05	PAYMENT	

25854 rows × 8 columns

In [73]:
```
clients.shape
# Has 1287(rows) x 3(Columns)
```

Out[73]: (1287, 3)

In [75]:
```
payments.shape
# Has 25559(Rows) x 6 (Columns)
```

Out[75]: (25559, 6)

In [86]:
```
clients.merge(payments, on="client_id", how="right", validate="one_to_many")
```

```
---------------------------------------------------------------------------
MergeError                                Traceback (most recent call last)
Cell In[86], line 1
----> 1 clients.merge(payments, on="client_id", how="right", validate="one_to_many")

File ~\AppData\Local\Programs\Python\Python311\Lib\site-packages\pandas\core\frame.py:10093, in DataFrame.merge(self, right, how, on, left_on, right_on, left_index, right_index, sort, suffixes, copy, indicator, validate)
  10074 @Substitution("")
  10075 @Appender(_merge_doc, indents=2)
  10076 def merge(
   (...)
  10089     validate: str | None = None,
  10090 ) -> DataFrame:
  10091     from pandas.core.reshape.merge import merge
> 10093     return merge(
  10094         self,
  10095         right,
  10096         how=how,
  10097         on=on,
  10098         left_on=left_on,
  10099         right_on=right_on,
  10100         left_index=left_index,
  10101         right_index=right_index,
  10102         sort=sort,
  10103         suffixes=suffixes,
  10104         copy=copy,
  10105         indicator=indicator,
  10106         validate=validate,
  10107     )

File ~\AppData\Local\Programs\Python\Python311\Lib\site-packages\pandas\core\reshape\merge.py:110, in merge(left, right, how, on, left_on, right_on, left_index, right_index, sort, suffixes, copy, indicator, validate)
     93 @Substitution("\nleft : DataFrame or named Series")
     94 @Appender(_merge_doc, indents=0)
     95 def merge(
   (...)
    108     validate: str | None = None,
    109 ) -> DataFrame:
--> 110     op = _MergeOperation(
    111         left,
    112         right,
    113         how=how,
    114         on=on,
    115         left_on=left_on,
    116         right_on=right_on,
    117         left_index=left_index,
    118         right_index=right_index,
    119         sort=sort,
    120         suffixes=suffixes,
    121         indicator=indicator,
    122         validate=validate,
    123     )
    124     return op.get_result(copy=copy)

File ~\AppData\Local\Programs\Python\Python311\Lib\site-packages\pandas\core\reshape\merge.py:713, in _MergeOperation.__init__(self, left, right, how, on, left_on, right_on, axis, left_index, right_index, sort, suffixes, indicator, valid
```

```
ate)
    709 # If argument passed to validate,
    710 # check if columns specified as unique
    711 # are in fact unique.
    712 if validate is not None:
--> 713     self._validate(validate)

File ~\AppData\Local\Programs\Python\Python311\Lib\site-packages\pandas\core\reshape\merge.py:1525, in _MergeOperation._validate(self, validate)
   1523 elif validate in ["one_to_many", "1:m"]:
   1524     if not left_unique:
-> 1525         raise MergeError(
   1526             "Merge keys are not unique in left dataset; not a one-to-many merge"
   1527         )
   1529 elif validate in ["many_to_one", "m:1"]:
   1530     if not right_unique:

MergeError: Merge keys are not unique in left dataset; not a one-to-many merge
```

The Above Merge() isn't reverting the exact rows that we need i.e. 25559 Because of multiple entries in clients['client_id'] and payments['client_id']

Hence, checking for exact issues with the data discreption

In [482…
```
#clients.groupby(['client_id','entity_type','entity_year_established'])['client_
```

In [100…
```
clients['client_id'].duplicated().sum()
```

Out[100]:
```
6
```

In [106…
```
clients.loc[clients['client_id'].duplicated()]
```

Out[106]:

	client_id	entity_type	entity_year_established
306	591	Australian Private Company	2007
350	473	Individual/Sole Trader	2008
816	591	Australian Private Company	2015
1115	165	Individual/Sole Trader	2006
1182	1262	Australian Private Company	2016
1275	797	Australian Private Company	2002

Shows an issue with anomalies regarding same client_id but changed either in

entity type or entity_type_established

While we know we can merge two dataframes based on Validate i.e. 'one_to_many' or 'many_to_many' or 'many_to_one'

In [143…
```
clients[clients.client_id.duplicated(keep=False)]
#clients.loc[~clients.client_id.unique()]
```

Out[143]:

	client_id	entity_type	entity_year_established
245	591	Australian Private Company	2013
306	591	Australian Private Company	2007
332	473	Australian Private Company	2016
350	473	Individual/Sole Trader	2008
401	165	Australian Private Company	2015
816	591	Australian Private Company	2015
1063	797	Discretionary Investment Trust	2016
1115	165	Individual/Sole Trader	2006
1172	1262	Australian Private Company	2005
1182	1262	Australian Private Company	2016
1275	797	Australian Private Company	2002

while using the same procedure to workout on payments['client_id']

In [145…
```
payments['client_id'].duplicated().sum()
```

Out[145]: 24278

In [148…
```
payments['client_id'].duplicated().count()
```

Out[148]: 25559

In [155…
```
payments.groupby(['contract_id','transaction_id','client_id'])['client_id'].cour
```

Out[155]:
```
contract_id  transaction_id  client_id
1            143             248          1
             371             248          1
             798             248          1
             1009            248          1
             1281            248          1
                                         ..
1641         25506           1194         1
1642         25507           37           1
             25508           37           1
1643         25509           299          1
             25510           299          1
Name: client_id, Length: 25559, dtype: int64
```

In [157…
```
payments.loc[payments['client_id'].duplicated()]
```

Out[157]:

	transaction_id	contract_id	client_id	transaction_date	payment_amt	payment_code
1	8485	927	1	1511716095	66.66	PAYMENT
2	13778	927	1	1519319303	66.66	PAYMENT
3	22768	927	1	1529863724	66.66	PAYMENT
4	15698	927	1	1521738504	66.66	PAYMENT
5	25167	927	1	1532282887	416.67	PAYMENT
...	...	...	...	...	...	...
25552	25385	1608	1279	1532282893	8333.34	PAYMENT
25554	25075	1603	1280	1532023764	1666.68	PAYMENT
25555	24711	1603	1280	1531764560	0.01	PAYMENT
25556	25076	1603	1280	1532023764	64.99	PAYMENT
25558	25131	1627	1281	1532282886	20000.05	PAYMENT

24278 rows × 6 columns

In [158…
```
payments.loc[payments['client_id'].unique()]
```

Out[158]:

	transaction_id	contract_id	client_id	transaction_date	payment_amt	payment_code
1	8485	927	1	1511716095	66.66	PAYMENT
2	13778	927	1	1519319303	66.66	PAYMENT
3	22768	927	1	1529863724	66.66	PAYMENT
4	15698	927	1	1521738504	66.66	PAYMENT
5	25167	927	1	1532282887	416.67	PAYMENT
...	...	...	...	...	...	...
1277	14316	874	64	1520183306	66.66	PAYMENT
1278	7217	874	64	1509901694	466.67	PAYMENT
1279	12412	874	64	1517764100	66.66	PAYMENT
1280	9480	897	65	1513184897	146.66	PAYMENT
1281	11266	897	65	1515949702	916.67	PAYMENT

1281 rows × 6 columns

The Above iteration Shows therea are multiple entries for client_id in payments['client_id'] which actually makes sense because of the several repayments.

Thus there are two approaches that can be done

1. Keeping the first entry of clients['client_id'] valid
2. Another way dropping the entries of duplicated values that has anomalies

Trying the second way of approach this time

In [191…

```
clients[clients.client_id.duplicated(keep=False)]
```

Out[191]:

	client_id	entity_type	entity_year_established
245	591	Australian Private Company	2013
306	591	Australian Private Company	2007
332	473	Australian Private Company	2016
350	473	Individual/Sole Trader	2008
401	165	Australian Private Company	2015
816	591	Australian Private Company	2015
1063	797	Discretionary Investment Trust	2016
1115	165	Individual/Sole Trader	2006
1172	1262	Australian Private Company	2005
1182	1262	Australian Private Company	2016
1275	797	Australian Private Company	2002

In [200…
```
clients.loc[clients['client_id'].duplicated()].client_id.count()
```

Out[200]: 6

In [166…
```
q = pd.merge(payments,clients,on='client_id',how='left')
print(q.head(5))
print('==============================\
      **********\
      ================================')
print(q.shape)
```

```
   transaction_id  contract_id  client_id  transaction_date  payment_amt  \
0           20175          927          1        1527012511        66.66
1            8485          927          1        1511716095        66.66
2           13778          927          1        1519319303        66.66
3           22768          927          1        1529863724        66.66
4           15698          927          1        1521738504        66.66

  payment_code        entity_type  entity_year_established
0      PAYMENT  Other Partnership                     2006
1      PAYMENT  Other Partnership                     2006
2      PAYMENT  Other Partnership                     2006
3      PAYMENT  Other Partnership                     2006
4      PAYMENT  Other Partnership                     2006
==============================      **********      ====================
============
(25854, 8)
```

In [190…
```
q.loc[q['client_id']==591].count()
```

Out[190]:
```
transaction_id             234
contract_id                234
client_id                  234
transaction_date           234
payment_amt                234
payment_code               234
entity_type                234
entity_year_established    234
dtype: int64
```

In [208…
```
q.loc[q['client_id'].isin([591, 473, 165, 591, 797, 1262])]
```

Out[208]:

	transaction_id	contract_id	client_id	transaction_date	payment_amt	payment_code
3255	18302	1280	165	1525025311	1716.67	PAYMENT
3256	18302	1280	165	1525025311	1716.67	PAYMENT
3257	17559	644	165	1524161307	1333.33	PAYMENT
3258	17559	644	165	1524161307	1333.33	PAYMENT
3259	20525	1280	165	1527444515	1666.67	PAYMENT
...	...	...	...	...	...	...
25779	24547	1584	1262	1531678160	833.34	PAYMENT
25780	23825	1439	1262	1530814127	2500.00	PAYMENT
25781	23825	1439	1262	1530814127	2500.00	PAYMENT
25782	21155	1439	1262	1528222125	0.01	PAYMENT
25783	21155	1439	1262	1528222125	0.01	PAYMENT

512 rows × 8 columns

There are total 512 Rows which has an anomaly with client_id and the following entity_type plus entity_year_established

So removing the duplicated values from the clients['Client_id']

In [219…
```
newclients = clients.drop_duplicates('client_id')
print(newclients.head(5))
print('****************************************')
print(newclients.shape)
print(payments.shape)
```

```
   client_id                entity_type  entity_year_established
0        786  Australian Private Company                     2002
1        230  Australian Private Company                     2008
2        282      Individual/Sole Trader                     2001
3        447  Australian Private Company                     2013
4        310      Individual/Sole Trader                     2015
****************************************
(1281, 3)
(25559, 6)
```

In [218…
```
newclients.loc[newclients.client_id.duplicated()].client_id.count()
```

Out[218]:
```
0
```

In [225…
```
merged = payments.merge(newclients, on='client_id', how='left', validate='many_t
print(merged.head(5))
print('********************************')
print(merged.shape)
```

```
   transaction_id  contract_id  client_id  transaction_date  payment_amt  \
0           20175          927          1        1527012511        66.66
1            8485          927          1        1511716095        66.66
2           13778          927          1        1519319303        66.66
3           22768          927          1        1529863724        66.66
4           15698          927          1        1521738504        66.66

  payment_code        entity_type  entity_year_established
0      PAYMENT  Other Partnership                     2006
1      PAYMENT  Other Partnership                     2006
2      PAYMENT  Other Partnership                     2006
3      PAYMENT  Other Partnership                     2006
4      PAYMENT  Other Partnership                     2006
********************************
(25559, 8)
```

Now the two files are merged with exact unique values of client_id that has been validated using validate()

In [227…
```
merged.head(5)
```

Out[227]:

	transaction_id	contract_id	client_id	transaction_date	payment_amt	payment_code	entity_
0	20175	927	1	1527012511	66.66	PAYMENT	C Partne
1	8485	927	1	1511716095	66.66	PAYMENT	C Partne
2	13778	927	1	1519319303	66.66	PAYMENT	C Partne
3	22768	927	1	1529863724	66.66	PAYMENT	C Partne
4	15698	927	1	1521738504	66.66	PAYMENT	C Partne

In [264…
```
date = pd.to_datetime(merged['transaction_date'], unit='s')
fdate = date.rename('date')
```

In [272…
```
final = pd.concat([merged, fdate], axis=1)
final
```

Out[272]:

	transaction_id	contract_id	client_id	transaction_date	payment_amt	payment_code	er
0	20175	927	1	1527012511	66.66	PAYMENT	P
1	8485	927	1	1511716095	66.66	PAYMENT	P
2	13778	927	1	1519319303	66.66	PAYMENT	P
3	22768	927	1	1529863724	66.66	PAYMENT	P
4	15698	927	1	1521738504	66.66	PAYMENT	P
...	...	...	...	...	...	...	
25554	25075	1603	1280	1532023764	1666.68	PAYMENT	
25555	24711	1603	1280	1531764560	0.01	PAYMENT	
25556	25076	1603	1280	1532023764	64.99	PAYMENT	
25557	25132	1627	1281	1532282886	0.01	PAYMENT	
25558	25131	1627	1281	1532282886	20000.05	PAYMENT	

25559 rows × 9 columns

In [274…

```
print(final.shape)
print('****************************************')
final.describe
```

```
(25559, 9)
****************************************
```

Out[274]:
```
<bound method NDFrame.describe of        transaction_id  contract_id  client_id
transaction_date  payment_amt  \
0                20175          927          1        1527012511        66.66
1                 8485          927          1        1511716095        66.66
2                13778          927          1        1519319303        66.66
3                22768          927          1        1529863724        66.66
4                15698          927          1        1521738504        66.66
...                ...          ...        ...               ...          ...
25554            25075         1603       1280        1532023764      1666.68
25555            24711         1603       1280        1531764560         0.01
25556            25076         1603       1280        1532023764        64.99
25557            25132         1627       1281        1532282886         0.01
25558            25131         1627       1281        1532282886     20000.05

      payment_code                 entity_type  entity_year_established  \
0          PAYMENT           Other Partnership                     2006
1          PAYMENT           Other Partnership                     2006
2          PAYMENT           Other Partnership                     2006
3          PAYMENT           Other Partnership                     2006
4          PAYMENT           Other Partnership                     2006
...            ...                         ...                      ...
25554      PAYMENT  Australian Private Company                     2016
25555      PAYMENT  Australian Private Company                     2016
25556      PAYMENT  Australian Private Company                     2016
25557      PAYMENT  Australian Private Company                     2012
25558      PAYMENT  Australian Private Company                     2012

                    date
0     2018-05-22 18:08:31
1     2017-11-26 17:08:15
2     2018-02-22 17:08:23
3     2018-06-24 18:08:44
4     2018-03-22 17:08:24
...                   ...
25554 2018-07-19 18:09:24
25555 2018-07-16 18:09:20
25556 2018-07-19 18:09:24
25557 2018-07-22 18:08:06
25558 2018-07-22 18:08:06

[25559 rows x 9 columns]>
```

The merged file is ready after appending the converted date and removing duplicates

Now, getting insights on the default payments.

Using groupby() method to derive the right answers

Getting values on entity_year Wise Defaults

In [277…
```
final.head(2)
```

Out[277]:

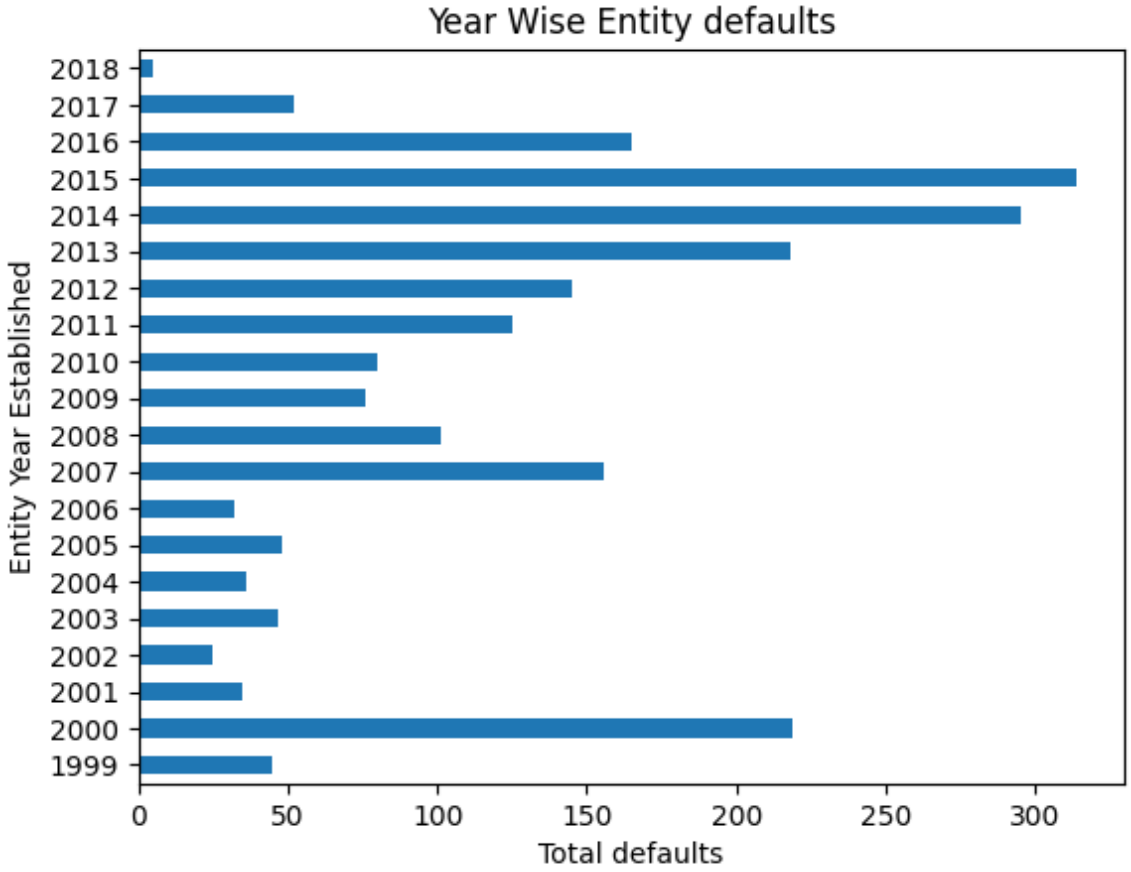
	transaction_id	contract_id	client_id	transaction_date	payment_amt	payment_code	entity_
0	20175	927	1	1527012511	66.66	PAYMENT	C Partne
1	8485	927	1	1511716095	66.66	PAYMENT	C Partne

In [280…
```
a = final.groupby(['entity_year_established','payment_code'])['payment_code'].cc
b = a.xs('DEFAULT', level='payment_code', axis=0)
b
```

Out[280]:
```
entity_year_established
1999     45
2000    219
2001     35
2002     25
2003     47
2004     36
2005     48
2006     32
2007    156
2008    101
2009     76
2010     80
2011    125
2012    145
2013    218
2014    295
2015    314
2016    165
2017     52
2018      5
Name: payment_code, dtype: int64
```

In [286…
```
b.plot(kind='barh', title='Year Wise Entity defaults', ylabel='Entity Year Estab
```

Out[286]:
```
<Axes: title={'center': 'Year Wise Entity defaults'}, xlabel='Total defaults',
ylabel='Entity Year Established'>
```



Maximum number of Defaults

In [329…

```
final.groupby(['entity_type']).count()
```

Out[329]:

entity_type	transaction_id	contract_id	client_id	transaction_date	payment_amt	payment_
Australian Private Company	14827	14827	14827	14827	14827	1
Australian Proprietary Company	4	4	4	4	4	
Australian Public Company	128	128	128	128	128	
Discretionary Investment Trust	124	124	124	124	124	
Discretionary Trading Trust	187	187	187	187	187	
Family Partnership	736	736	736	736	736	
Fixed Unit Trust	32	32	32	32	32	
Hybrid Trust	32	32	32	32	32	
Individual/Sole Trader	9354	9354	9354	9354	9354	
Other Partnership	135	135	135	135	135	

In [288…
```
c = final.groupby(['entity_type','payment_code'])['payment_code'].count()
d = c.xs('DEFAULT', level='payment_code', axis=0)
d
```

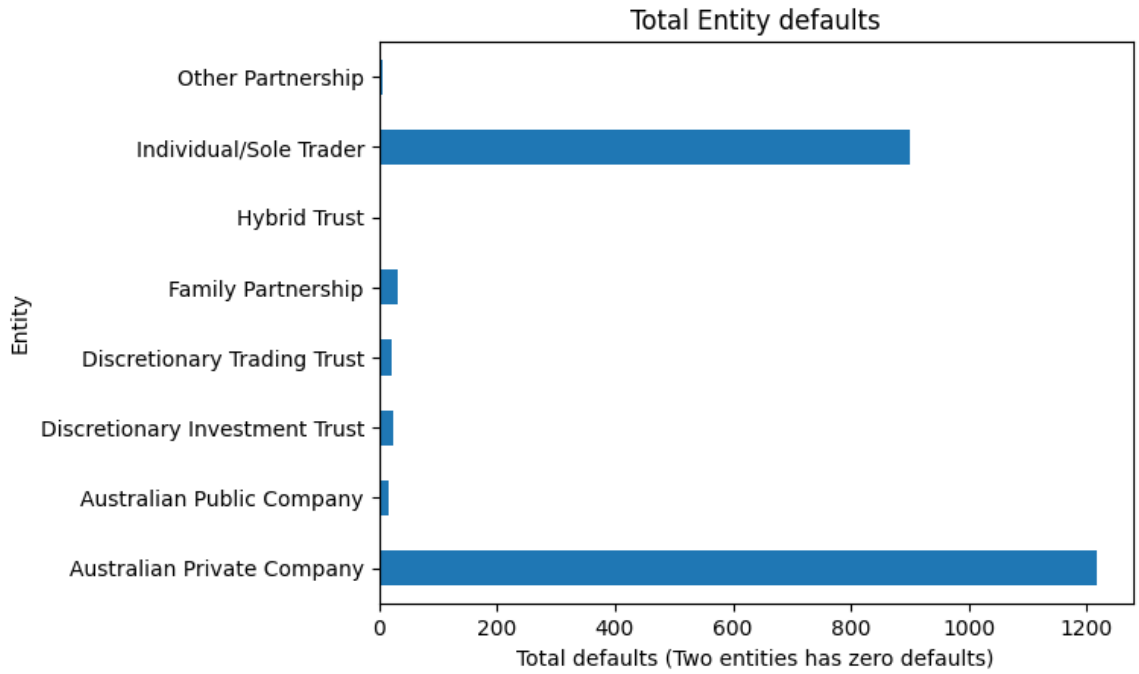
Out[288]:
```
entity_type
Australian Private Company        1217
Australian Public Company           16
Discretionary Investment Trust      23
Discretionary Trading Trust         21
Family Partnership                  33
Hybrid Trust                         2
Individual/Sole Trader             901
Other Partnership                    6
Name: payment_code, dtype: int64
```

In [301…
```
final.loc[final.payment_code=='DEFAULT'].entity_type.count()
```

Out[301]:
```
2219
```

In [323…
```
d.plot(kind='barh', title='Total Entity defaults', ylabel='Entity', xlabel='Tota
(Two entities has zero defaults)')
```

Out[323]:
```
<Axes: title={'center': 'Total Entity defaults'}, xlabel='Total defaults (Two e
ntities has zero defaults)', ylabel='Entity'>
```

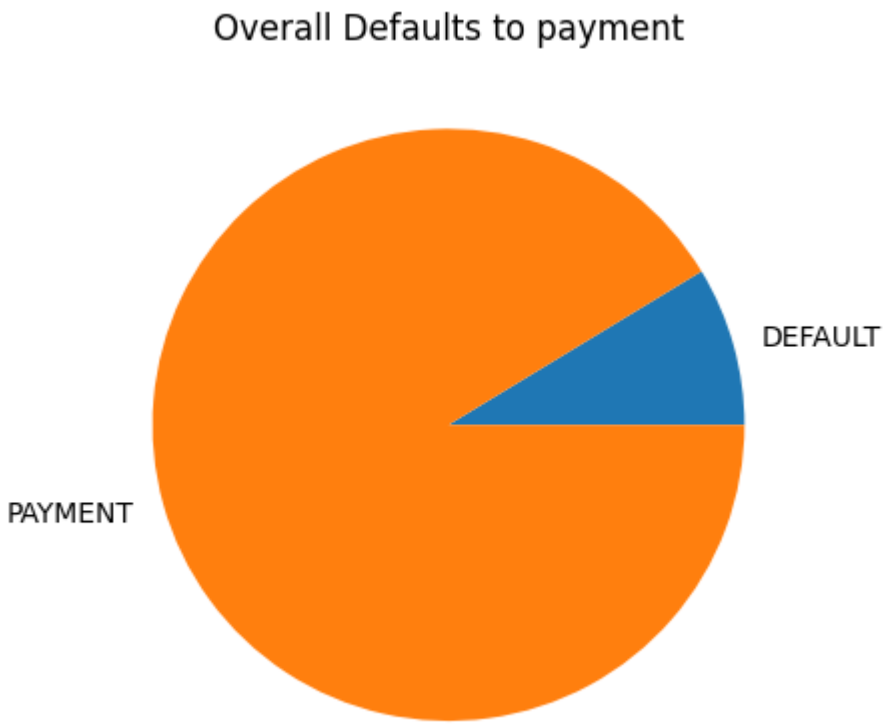


In [346…
```
e = final.groupby(['payment_code'])['entity_type'].count()
e
```

Out[346]:
```
payment_code
DEFAULT     2219
PAYMENT    23340
Name: entity_type, dtype: int64
```

In [350…
```
e.plot(kind='pie', title='Overall Defaults to payment', ylabel='')
```

Out[350]:
```
<Axes: title={'center': 'Overall Defaults to payment'}>
```



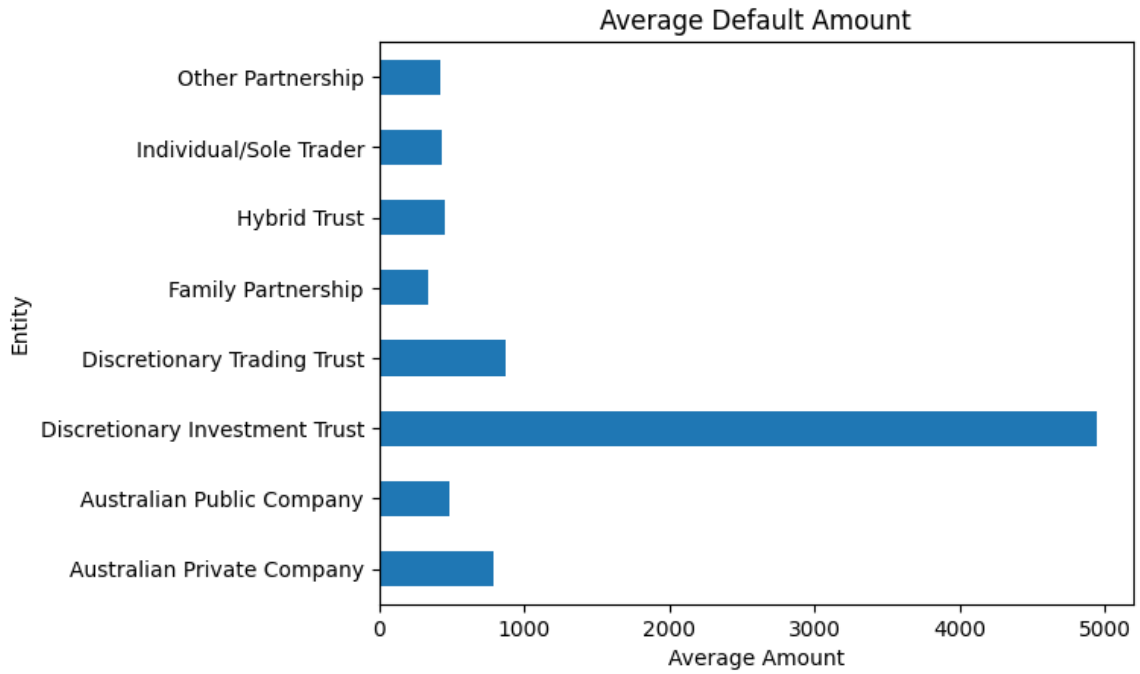
In [360…
```
f = final.groupby(['entity_type','payment_code'])['payment_amt'].mean()
g = f.xs('DEFAULT', level='payment_code', axis=0)
```

```
g
#y.xs('Australian Private Company', level='entity_type', axis=0)
#y['entity_type']=='Australian Private Company'
```

Out[360]:
```
entity_type
Australian Private Company         785.943361
Australian Public Company          485.676250
Discretionary Investment Trust    4946.242174
Discretionary Trading Trust        877.671905
Family Partnership                 341.464848
Hybrid Trust                       450.000000
Individual/Sole Trader             428.108923
Other Partnership                  420.988333
Name: payment_amt, dtype: float64
```

In [367…
```
g.plot(kind='barh', title='Average Default Amount', xlabel='Average Amount', yla
```

Out[367]:
```
<Axes: title={'center': 'Average Default Amount'}, xlabel='Average Amount', yla
bel='Entity'>
```



In [377…
```
h = final.groupby(['entity_type', 'payment_code','entity_year_established'])['pa
i = h.xs('DEFAULT', level='payment_code', axis=0)
i
```

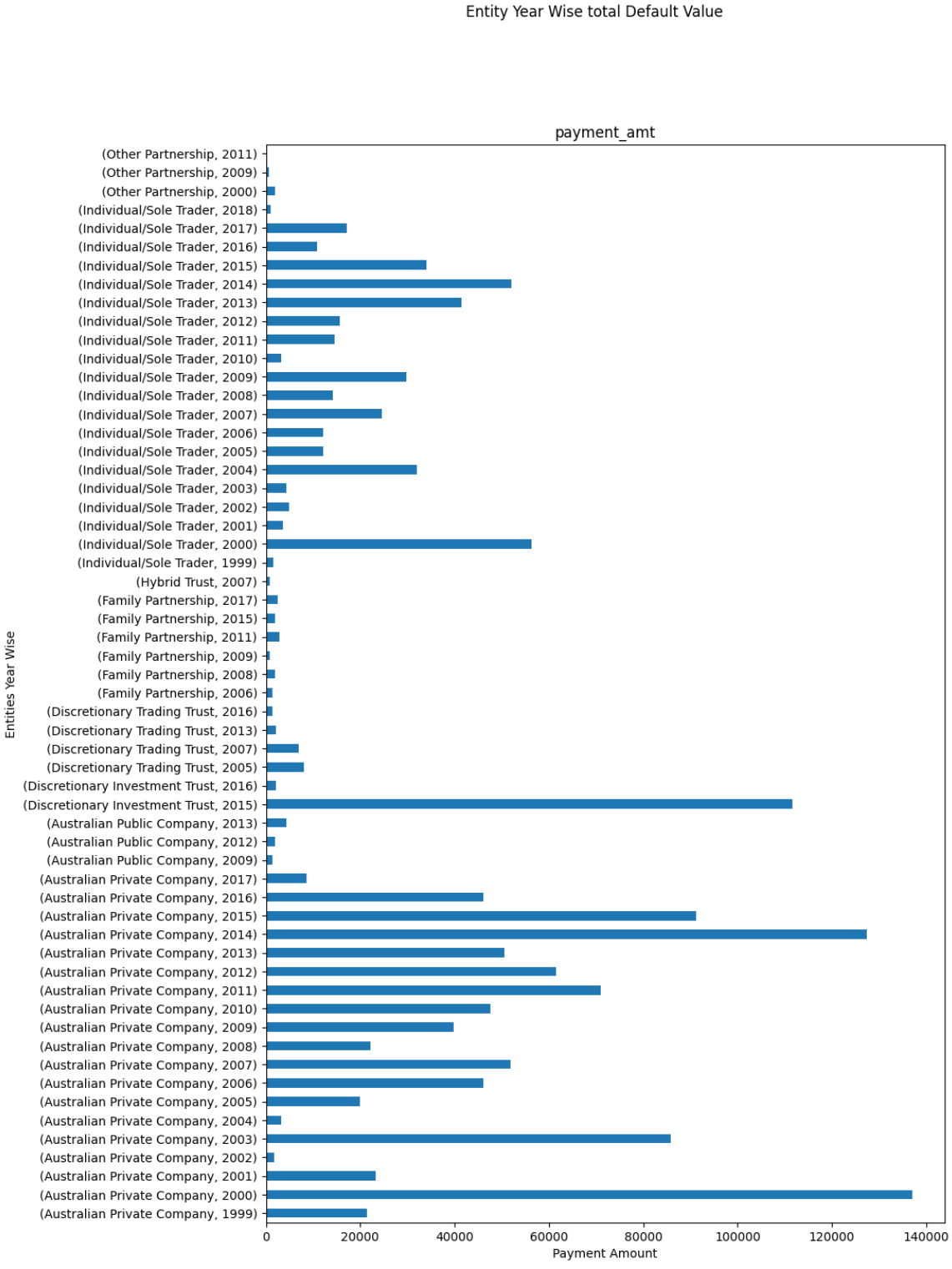
Out[377]:

entity_type	entity_year_established	
Australian Private Company	1999	21351.07
	2000	136985.78
	2001	23371.99
	2002	1740.00
	2003	85910.28
	2004	3182.50
	2005	19858.31
	2006	46166.64
	2007	51877.42
	2008	22137.73
	2009	39768.37
	2010	47676.98
	2011	70895.67
	2012	61530.72
	2013	50655.55
	2014	127432.77
	2015	91243.19
	2016	46091.49
	2017	8616.61
Australian Public Company	2009	1437.48
	2012	1933.33
	2013	4400.01
Discretionary Investment Trust	2015	111620.00
	2016	2143.57
Discretionary Trading Trust	2005	8000.00
	2007	6991.66
	2013	2139.45
	2016	1300.00
Family Partnership	2006	1353.32
	2008	1884.99
	2009	746.66
	2011	2836.72
	2015	1933.33
	2017	2513.32
Hybrid Trust	2007	900.00
Individual/Sole Trader	1999	1620.83
	2000	56260.61
	2001	3531.63
	2002	4977.78
	2003	4411.56
	2004	32068.23
	2005	12133.28
	2006	12095.53
	2007	24656.55
	2008	14197.51
	2009	29709.08
	2010	3154.97
	2011	14607.31
	2012	15703.28
	2013	41481.63
	2014	52072.94
	2015	33974.87
	2016	10888.75
	2017	17229.56
	2018	950.24
Other Partnership	2000	1933.32
	2009	565.26
	2011	27.35

Name: payment_amt, dtype: float64

In [412…
```
i.plot(kind='barh', subplots=True, figsize=(10,16), title='Entity Year Wise tota
```

Out[412]:
```
array([<Axes: title={'center': 'payment_amt'}, xlabel='Payment Amount', ylabel
='Entities Year Wise'>],
      dtype=object)
```



Entity Grouped by Total Contract_ID and Date

In [431…
```
j = final.groupby(['entity_type','payment_code', 'date'])['payment_amt'].sum()
k = j.xs('DEFAULT', level='payment_code', axis=0)
```

```
k
```

Out[431]:
```
entity_type                 date
Australian Private Company  2017-07-03 18:08:05    2438.39
                            2017-07-04 18:08:07    4748.33
                            2017-07-05 18:08:08    3122.37
                            2017-07-06 18:08:10    5000.00
                            2017-07-10 18:08:14      30.46
                                                     ...
Individual/Sole Trader      2018-07-24 18:08:08    3193.33
Other Partnership           2017-08-03 18:08:48      27.35
                            2017-08-09 18:08:56     565.26
                            2017-10-26 17:08:13     966.66
                            2017-11-28 17:08:13     966.66
Name: payment_amt, Length: 611, dtype: float64
```

In [436…
```
l = final.groupby(['entity_type'])['payment_code'].count()
#m = l.xs('DEFAULT', level='payment_code', axis=0)
l
```

Out[436]:
```
entity_type
Australian Private Company        14827
Australian Proprietary Company        4
Australian Public Company           128
Discretionary Investment Trust      124
Discretionary Trading Trust         187
Family Partnership                  736
Fixed Unit Trust                     32
Hybrid Trust                         32
Individual/Sole Trader             9354
Other Partnership                   135
Name: payment_code, dtype: int64
```

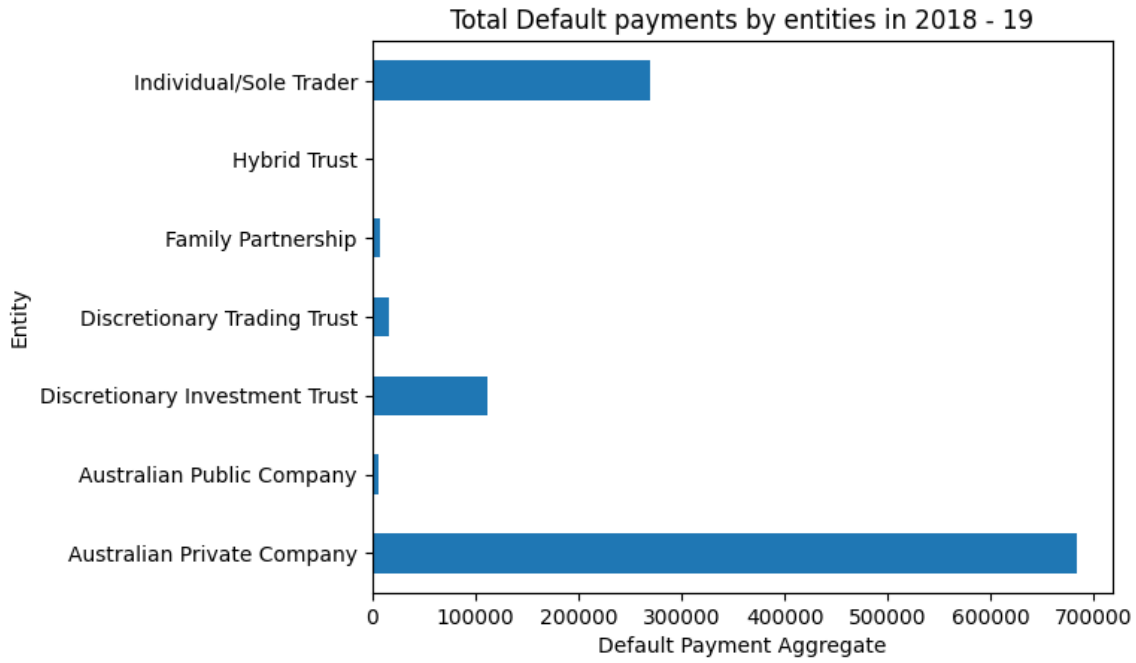
In [461…
```
k17 = final[(final['date'] > '2017-01-01 00:00:00') & (final['date'] < '2017-12-
print('Total Sum of payment made during 2017-2018 is =', k17)
k18 = final[(final['date'] > '2018-01-01 00:00:00') & (final['date'] < '2018-12-
print('Total Sum of payment made during 2018-2019 is =', k18)
k19 = final[(final['date'] > '2019-01-01 00:00:00') & (final['date'] < '2019-12-
print('Total Sum of payment made during 2019-2020 is =', k19)
k16 = final[(final['date'] > '2016-01-01 00:00:00') & (final['date'] < '2016-12-
print('Total Sum of payment made during 2016-2017 is =', k16)
```

```
Total Sum of payment made during 2017-2018 is = 9337016.69
Total Sum of payment made during 2018-2019 is = 21882169.310000002
Total Sum of payment made during 2019-2020 is = 0.0
Total Sum of payment made during 2016-2017 is = 0.0
```

In [472…
```
m = final[(final['date'] > '2018-01-01 00:00:00') & (final['date'] < '2018-12-31
n = m.groupby(['entity_type','payment_code'])['payment_amt'].sum()
o = n.xs('DEFAULT', level='payment_code', axis=0)
print(o)
o.plot(kind='barh', title='Total Default payments by entities in 2018 - 19', yla
```

```
entity_type
Australian Private Company        683505.66
Australian Public Company           6333.34
Discretionary Investment Trust    111891.57
Discretionary Trading Trust        15800.64
Family Partnership                  7684.96
Hybrid Trust                         900.00
Individual/Sole Trader            269148.13
Name: payment_amt, dtype: float64
```

Out[472]: <Axes: title={'center': 'Total Default payments by entities in 2018 - 19'}, xlabel='Default Payment Aggregate', ylabel='Entity'>

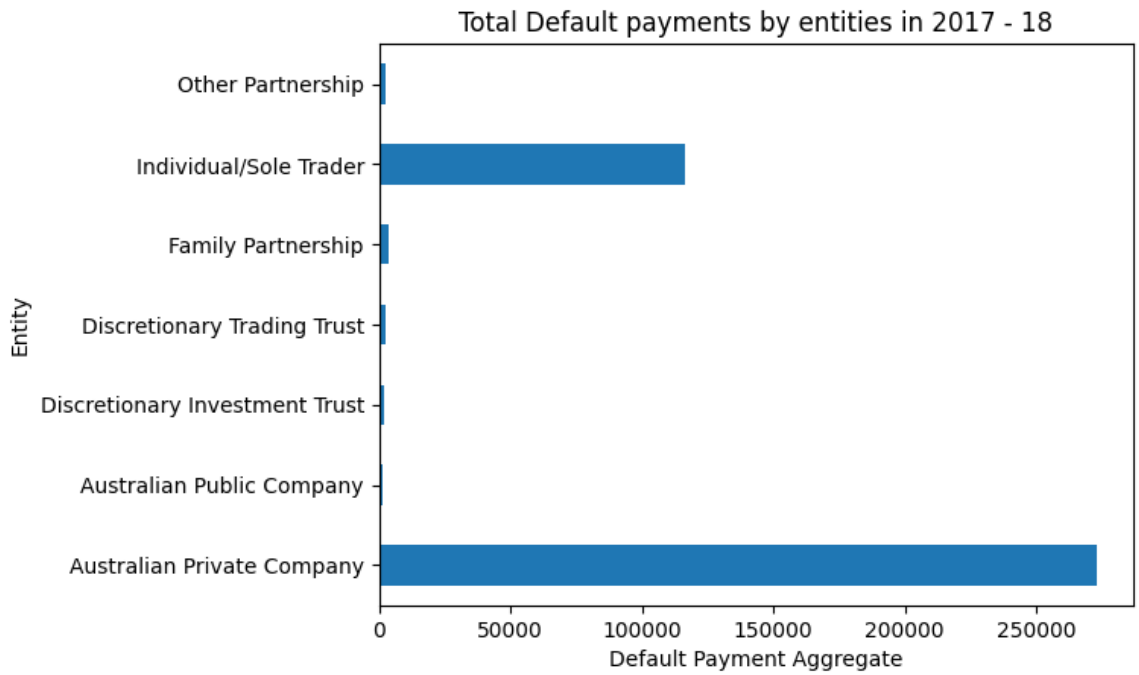


In [473…

```
p = final[(final['date'] > '2017-01-01 00:00:00') & (final['date'] < '2017-12-31
q = p.groupby(['entity_type','payment_code'])['payment_amt'].sum()
r = q.xs('DEFAULT', level='payment_code', axis=0)
print(r)
r.plot(kind='barh', title='Total Default payments by entities in 2017 - 18', yla
```

```
entity_type
Australian Private Company        272987.41
Australian Public Company           1437.48
Discretionary Investment Trust      1872.00
Discretionary Trading Trust         2630.47
Family Partnership                  3583.38
Individual/Sole Trader            116578.01
Other Partnership                   2525.93
Name: payment_amt, dtype: float64
```

Out[473]: <Axes: title={'center': 'Total Default payments by entities in 2017 - 18'}, xlabel='Default Payment Aggregate', ylabel='Entity'>



In [481…

```
print('Total SUM of default payment made in 2018-2019 is =', o.sum())
print('Total SUM of default payment made in 2017-2018 is =', r.sum())
print('Total SUM of default payment made in OVERALL is =', o.sum()+r.sum())
```

```
Total SUM of default payment made in 2018-2019 is = 1095264.3
Total SUM of default payment made in 2017-2018 is = 401614.67999999993
Total SUM of default payment made in OVERALL is = 1496878.98
```

In [485…

```
DescriptiveReport = ProfileReport(final)
DescriptiveReport.to_file('FinalER-Report.html')
```

```
Summarize dataset: 100%|██████████████████████████████████████████████████| 5
4/54 [00:10<00:00,  4.97it/s, Completed]
Generate report structure: 100%|
██████████| 1/1 [00:05<00:00,  5.81s/it]
Render HTML: 100%|
██████████| 1/1 [00:02<00:00,  2.48s/it]
Export report to file: 100%|
█████████| 1/1 [00:00<00:00, 124.58it/s]
```