```
import numpy, scipy.stats, sklearn.isotonic, tabulate
```

```
# manual_bin

def manual_bin1(x, y, cuts):
  """
  base
  """
  _x = [_ for _ in x]
  _y = [_ for _ in y]
  _c = sorted([_ for _ in set(cuts)] + [numpy.NINF, numpy.PINF])
  _g = numpy.searchsorted(_c, _x).tolist()

  _l1 = list(dict(zip(['g', 'x', 'y'], _)) for _ in zip(_g, _x, _y))
  _l2 = zip(set(_g), [[l for l in _l1 if l["g"] == g] for g in set(_g)])

  return(sorted([dict(zip(["bin", "freq", "miss", "bads", "minx", "maxx"],
                          [_1,
                           len(_2),
                           0,
                           sum([_["y"] for _ in _2]),
                           min([_["x"] for _ in _2]),
                           max([_["x"] for _ in _2])])) for _1, _2 in _l2],
                key = lambda x: x["bin"]))


def manual_bin2(x, y, cuts):
    _x = [_ for _ in x]
    _y = [_ for _ in y]
    _c = sorted([_ for _ in set(cuts)] + [numpy.NINF, numpy.PINF])
    _g = numpy.searchsorted(_c, _x).tolist()

    """
    zip g,x,y and group and done all.
    """
    _d1 = {}
    for __g, __x, __y in zip(_g, _x, _y):
        if __g not in _d1:
            _d1[__g] = {"freq": 1, "bads": __y, "minx": __x, "maxx": __x}
        else:
            _d1[__g] = {
                "freq": _d1[__g]["freq"]+1,
                "bads": _d1[__g]["bads"]+__y,
                "minx": min(_d1[__g]["minx"], __x),
                "maxx": max(_d1[__g]["maxx"], __x)
            }

    return sorted([dict({"bin": _1, "miss": 0,}, **_2) for _1, _2 in _d1.items()], key = lambda x: x["bin"])
```

```

##############################################################################################################

def qtl_bin(x, y, func):
  _data = [_ for _ in zip(x, y, ~numpy.isnan(x))]
  _freq = len(_data)
  _bads = sum([_[1] for _ in _data])

  _x = [_[0] for _ in _data if _[2] == 1]
  _y = [_[1] for _ in _data if _[2] == 1]

  _n = numpy.arange(2, max(3, min(50, len(numpy.unique(_x)) - 1)))
  _p = numpy.unique([qcut(_x, _) for _ in _n])


  """
  manual_bin to func
  """
  _l1 = [[_, func(_x, _y, _)] for _ in _p]
  # ...
```

Test...

```
import sas7bdat
import pandas as pd
df = sas7bdat.SAS7BDAT("accepts.sas7bdat").to_data_frame()
```

```
df_s = df[["purch_price", "bad"]]
print(df_s.shape)

utl = df_s.purch_price
bad = df_s.bad

%time utl_bin = qtl_bin(utl, bad, manual_bin1)
view_bin(utl_bin)
print()

%time utl_bin = qtl_bin(utl, bad, manual_bin2)
view_bin(utl_bin)
print()
```

```
(5837, 2)
CPU times: user 841 ms, sys: 3.47 ms, total: 845 ms
Wall time: 845 ms
|   bin |   freq |   miss |   bads |   rate |     woe |     iv | rule                                    |
|-------+--------+--------+--------+--------+---------+--------+-----------------------------------------|
|     1 |   1171 |      0 |    283 | 0.2417 |  0.2124 | 0.0096 | $X$ <= 11300.0                          |
|     2 |   1164 |      0 |    264 | 0.2268 |  0.1295 | 0.0035 | ($X$ > 11300.0) and ($X$ <= 15800.48)   |
|     3 |   1171 |      0 |    227 | 0.1939 | -0.0692 | 0.0009 | ($X$ > 15800.48) and ($X$ <= 20500.0)   |
|     4 |   1163 |      0 |    213 | 0.1831 | -0.1392 | 0.0037 | ($X$ > 20500.0) and ($X$ <= 26130.0)    |
|     5 |   1168 |      0 |    209 | 0.1789 | -0.1676 | 0.0053 | $X$ > 26130.0                           |

CPU times: user 511 ms, sys: 1.57 ms, total: 512 ms
Wall time: 522 ms
|   bin |   freq |   miss |   bads |   rate |     woe |     iv | rule                                    |
|-------+--------+--------+--------+--------+---------+--------+-----------------------------------------|
|     1 |   1171 |      0 |    283 | 0.2417 |  0.2124 | 0.0096 | $X$ <= 11300.0                          |
|     2 |   1164 |      0 |    264 | 0.2268 |  0.1295 | 0.0035 | ($X$ > 11300.0) and ($X$ <= 15800.48)   |
|     3 |   1171 |      0 |    227 | 0.1939 | -0.0692 | 0.0009 | ($X$ > 15800.48) and ($X$ <= 20500.0)   |
|     4 |   1163 |      0 |    213 | 0.1831 | -0.1392 | 0.0037 | ($X$ > 20500.0) and ($X$ <= 26130.0)    |
|     5 |   1168 |      0 |    209 | 0.1789 | -0.1676 | 0.0053 | $X$ > 26130.0                           |
```

Test big data...

```
df_b = pd.concat([df[["purch_price", "bad"]] for i in range(200)])
print(df_b.shape)

utl = df_b.purch_price
bad = df_b.bad

%time utl_bin = qtl_bin(utl, bad, manual_bin1)
view_bin(utl_bin)
print()

%time utl_bin = qtl_bin(utl, bad, manual_bin2)
view_bin(utl_bin)
print()
```

```
(1167400, 2)
CPU times: user 3min 42s, sys: 4.79 s, total: 3min 47s
Wall time: 3min 49s
|   bin |   freq |   miss |   bads |   rate |     woe |     iv | rule                                    |
|-------+--------+--------+--------+--------+---------+--------+-----------------------------------------|
|     1 | 234200 |      0 |  56600 | 0.2417 |  0.2124 | 0.0096 | $X$ <= 11300.0                          |
|     2 | 232800 |      0 |  52800 | 0.2268 |  0.1295 | 0.0035 | ($X$ > 11300.0) and ($X$ <= 15800.48)   |
|     3 | 234200 |      0 |  45400 | 0.1939 | -0.0692 | 0.0009 | ($X$ > 15800.48) and ($X$ <= 20500.0)   |
|     4 | 232800 |      0 |  42600 | 0.183  | -0.1403 | 0.0038 | ($X$ > 20500.0) and ($X$ <= 26134.95)   |
|     5 | 233400 |      0 |  41800 | 0.1791 | -0.1666 | 0.0053 | $X$ > 26134.95                          |

CPU times: user 1min 49s, sys: 2.38 s, total: 1min 51s
Wall time: 1min 53s
|   bin |   freq |   miss |   bads |   rate |     woe |     iv | rule                                    |
|-------+--------+--------+--------+--------+---------+--------+-----------------------------------------|
|     1 | 234200 |      0 |  56600 | 0.2417 |  0.2124 | 0.0096 | $X$ <= 11300.0                          |
|     2 | 232800 |      0 |  52800 | 0.2268 |  0.1295 | 0.0035 | ($X$ > 11300.0) and ($X$ <= 15800.48)   |
|     3 | 234200 |      0 |  45400 | 0.1939 | -0.0692 | 0.0009 | ($X$ > 15800.48) and ($X$ <= 20500.0)   |
|     4 | 232800 |      0 |  42600 | 0.183  | -0.1403 | 0.0038 | ($X$ > 20500.0) and ($X$ <= 26134.95)   |
|     5 | 233400 |      0 |  41800 | 0.1791 | -0.1666 | 0.0053 | $X$ > 26134.95                          |
```

Test memory used...

```
# ! /Users/ivan/Desktop/ALL/Soft/python3/bin/pip install memory_profiler psutil
```

```
! /Users/ivan/Desktop/ALL/Soft/python3/bin/python -m memory_profiler pymob.py 1
```

```
Filename: pymob.py

Line #    Mem usage    Increment   Line Contents
================================================
   518  135.121 MiB  135.121 MiB   @profile
   519                             def f():
   520  135.121 MiB    0.000 MiB       if n == "1":
   521  935.980 MiB  800.859 MiB           utl_bin = qtl_bin(utl, bad, manual_bin1)
   522  935.980 MiB    0.000 MiB       if n == "2":
   523                                   utl_bin = qtl_bin(utl, bad, manual_bin2)
```

```
! /Users/ivan/Desktop/ALL/Soft/python3/bin/python -m memory_profiler pymob.py 2
```

```
Filename: pymob.py

Line #    Mem usage    Increment   Line Contents
================================================
   518  135.730 MiB  135.730 MiB   @profile
   519                             def f():
   520  135.730 MiB    0.000 MiB       if n == "1":
   521                                   utl_bin = qtl_bin(utl, bad, manual_bin1)
   522  135.730 MiB    0.000 MiB       if n == "2":
   523  358.324 MiB  222.594 MiB           utl_bin = qtl_bin(utl, bad, manual_bin2)
```