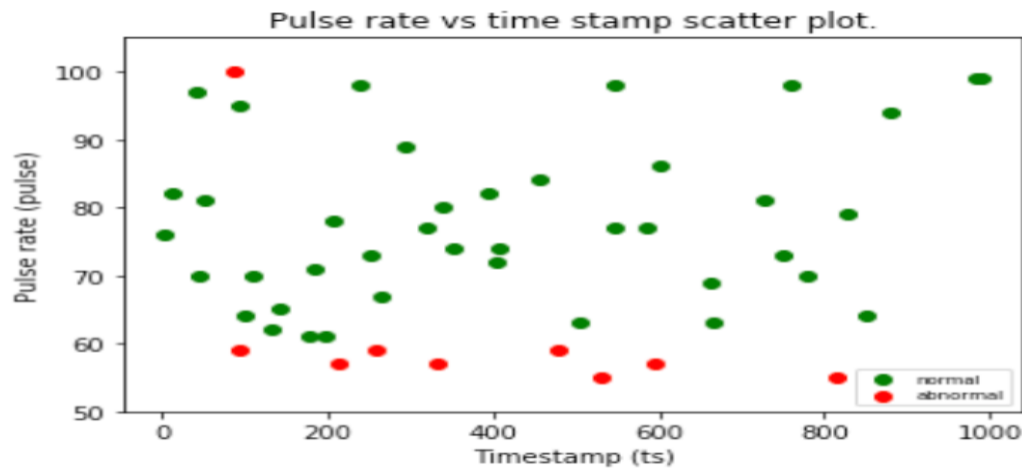


**Technical Annexe PWD CW2:****Phase 1 Function output:**

```
mydict=myHealthcare(1000)
print(mydict)
```

{0: [36, 80, 55, 120, 16, 94, 7.3], 1: [39, 61, 56, 121, 11, 93, 7.2], 2: [37, 67, 64, 120, 13, 99, 7.5], 3: [36, 72, 76, 120, 11, 93, 7.2], 4: [39, 83, 82, 120, 15, 99, 7.2], 5: [37, 99, 73, 120, 16, 98, 7.3], 6: [38, 98, 67, 121, 17, 96, 7.3], 7: [39, 71, 71, 121, 12, 93, 7.2], 8: [36, 84, 55, 121, 14, 94, 7.2], 9: [38, 92, 57, 121, 15, 97, 7.4], 10: [38, 96, 67, 121, 11, 95, 7.4], 11: [36, 71, 82, 120, 13, 94, 7.3], 12: [37, 69, 65, 121, 17, 96, 7.2], 13: [39, 98, 68, 121, 17, 94, 7.1], 14: [37, 99, 81, 120, 17, 99, 7.5], 15: [39, 87, 62, 121, 16, 100, 7.3], 16: [39, 68, 88, 121, 14, 97, 7.3], 17: [39, 78, 78, 120, 11, 99, 7.5], 18: [38, 87, 57, 120, 15, 93, 7.4], 19: [36, 67, 74, 120, 11, 93, 7.5], 20: [36, 70, 61, 120, 13, 93, 7.2], 21: [39, 82, 86, 120, 11, 93, 7.5], 22: [36, 84, 100, 120, 12, 100, 7.5], 23: [39, 56, 71, 120, 15, 100, 7.5], 24: [38, 60, 98, 121, 15, 98, 7.3], 25: [39, 98, 100, 121, 15, 98, 7.3], 26: [39, 63, 83, 120, 15, 98, 7.5], 27: [38, 65, 99, 121, 15, 100, 7.2], 28: [39, 64, 92, 121, 16, 94, 7.3], 29: [36, 69, 80, 121, 17, 93, 7.2], 30: [37, 73, 93, 120, 16, 95, 7.1], 31: [39, 63, 87, 121, 12, 95, 7.2], 32: [37, 80, 72, 120, 17, 99, 7.6], 33: [38, 89, 68, 120, 16, 98, 7.3], 34: [39, 65, 96, 121, 15, 97, 7.5], 35: [39, 90, 72, 120, 12, 97, 7.4], 36: [38, 85, 69, 120, 17, 99, 7.2], 37: [39, 61, 98, 120, 11, 98, 7.4], 38: [38, 87, 61, 121, 15, 98, 7.5], 39: [37, 98, 71, 120, 13, 96, 7.6], 40: [36, 65, 86, 121, 12, 94, 7.2], 41: [38, 77, 57, 120, 17, 99, 7.3], 42: [36, 80, 97, 121, 12, 100, 7.3], 43: [37, 92, 78, 120, 16, 93, 7.2], 44: [36, 96, 72, 121, 14, 96, 7.5], 45: [36, 74, 70, 120, 11, 93, 7.4], 46: [39, 60, 76, 121, 15, 93, 7.1], 47: [36, 79, 59, 121, 14, 97, 7.6], 48: [36, 83, 64, 120, 14, 97, 7.3], 49: [39, 73, 61, 121, 14, 93, 7.4], 50: [38, 61, 97, 120, 13, 100, 7.3], 51: [38, 95, 91, 121, 12, 100, 7.5], 52: [37, 98, 81, 120, 14, 94, 7.3], 53: [36, 81, 80, 120, 11, 96, 7.6], 54: [39, 85, 61, 121, 14, 93, 7.5], 55: [39, 55, 100, 121, 16, 99, 7.1], 56: [39, 78, 55, 120, 14, 98, 7.2], 57: [39, 100, 89, 120, 13, 100, 7.2], 58: [38, 98, 84, 121, 12, 95, 7.3], 59: [38, 90, 84, 121, 12, 94, 7.4], 60: [37, 90, 95, 120, 16, 94, 7.4], 61: [37, 90, 86, 120, 14, 97, 7.1], 62: [38, 80, 95, 121, 12, 99, 7.4], 63: [38, 93, 93, 120, 12, 96, 7.2], 64: [36, 65, 75, 120, 17, 96, 7.5], 65: [38, 85, 69, 120, 17, 99, 7.2], 66: [39, 61, 98, 120, 11, 98, 7.4], 67: [38, 87, 61, 121, 15, 98, 7.5], 68: [37, 98, 71, 120, 13, 96, 7.6], 69: [36, 65, 86, 121, 12, 94, 7.2], 70: [38, 77, 57, 120, 17, 99, 7.3], 71: [36, 80, 97, 121, 12, 100, 7.3], 72: [37, 92, 78, 120, 16, 93, 7.2], 73: [36, 96, 72, 121, 14, 96, 7.5], 74: [36, 74, 70, 120, 11, 93, 7.4], 75: [39, 60, 76, 121, 15, 93, 7.1], 76: [36, 79, 59, 121, 14, 97, 7.6], 77: [36, 83, 64, 120, 14, 97, 7.3], 78: [39, 73, 61, 121, 14, 93, 7.4], 79: [38, 61, 97, 120, 13, 100, 7.3], 80: [38, 95, 91, 121, 12, 100, 7.5], 81: [37, 98, 81, 120, 14, 94, 7.3], 82: [36, 81, 80, 120, 11, 96, 7.6], 83: [39, 85, 61, 121, 14, 93, 7.5], 84: [39, 55, 100, 121, 16, 99, 7.1], 85: [39, 78, 55, 120, 14, 98, 7.2], 86: [39, 100, 89, 120, 13, 100, 7.2], 87: [38, 98, 84, 121, 12, 95, 7.3], 88: [38, 90, 84, 121, 12, 94, 7.4], 89: [37, 90, 95, 120, 16, 94, 7.4], 90: [37, 90, 86, 120, 14, 97, 7.1], 91: [38, 80, 95, 121, 12, 99, 7.4], 92: [38, 93, 93, 120, 12, 96, 7.2], 93: [36, 65, 75, 120, 17, 96, 7.5], 94: [38, 85, 69, 120, 17, 99, 7.2], 95: [39, 61, 98, 120, 11, 98, 7.4], 96: [38, 87, 61, 121, 15, 98, 7.5], 97: [37, 98, 71, 120, 13, 96, 7.6], 98: [36, 65, 86, 121, 12, 94, 7.2], 99: [38, 77, 57, 120, 17, 99, 7.3], 100: [36, 80, 97, 121, 12, 100, 7.3], 101: [37, 92, 78, 120, 16, 93, 7.2], 102: [36, 96, 72, 121, 14, 96, 7.5], 103: [36, 74, 70, 120, 11, 93, 7.4], 104: [39, 60, 76, 121, 15, 93, 7.1], 105: [36, 79, 59, 121, 14, 97, 7.6], 106: [36, 83, 64, 120, 14, 97, 7.3], 107: [39, 73, 61, 121, 14, 93, 7.4], 108: [38, 61, 97, 120, 13, 100, 7.3], 109: [38, 95, 91, 121, 12, 100, 7.5], 110: [37, 98, 81, 120, 14, 94, 7.3], 111: [36, 81, 80, 120, 11, 96, 7.6], 112: [39, 85, 61, 121, 14, 93, 7.5], 113: [39, 55, 100, 121, 16, 99, 7.1], 114: [39, 78, 55, 120, 14, 98, 7.2], 115: [39, 100, 89, 120, 13, 100, 7.2], 116: [38, 98, 84, 121, 12, 95, 7.3], 117: [38, 90, 84, 121, 12, 94, 7.4], 118: [37, 90, 95, 120, 16, 94, 7.4], 119: [37, 90, 86, 120, 14, 97, 7.1], 120: [38, 80, 95, 121, 12, 99, 7.4], 121: [38, 93, 93, 120, 12, 96, 7.2], 122: [36, 65, 75, 120, 17, 96, 7.5], 123: [38, 85, 69, 120, 17, 99, 7.2], 124: [39, 61, 98, 120, 11, 98, 7.4], 125: [38, 87, 61, 121, 15, 98, 7.5], 126: [37, 98, 71, 120, 13, 96, 7.6], 127: [36, 65, 86, 121, 12, 94, 7.2], 128: [38, 77, 57, 120, 17, 99, 7.3], 129: [36, 80, 97, 121, 12, 100, 7.3], 130: [37, 92, 78, 120, 16, 93, 7.2], 131: [36, 96, 72, 121, 14, 96, 7.5], 132: [36, 74, 70, 120, 11, 93, 7.4], 133: [39, 60, 76, 121, 15, 93, 7.1], 134: [36, 79, 59, 121, 14, 97, 7.6], 135: [36, 83, 64, 120, 14, 97, 7.3], 136: [39, 73, 61, 121, 14, 93, 7.4], 137: [38, 61, 97, 120, 13, 100, 7.3], 138: [38, 95, 91, 121, 12, 100, 7.5], 139: [37, 98, 81, 120, 14, 94, 7.3], 140: [36, 81, 80, 120, 11, 96, 7.6], 141: [39, 85, 61, 121, 14, 93, 7.5], 142: [39, 55, 100, 121, 16, 99, 7.1], 143: [39, 78, 55, 120, 14, 98, 7.2], 144: [39, 100, 89, 120, 13, 100, 7.2], 145: [38, 98, 84, 121, 12, 95, 7.3], 146: [38, 90, 84, 121, 12, 94, 7.4], 147: [37, 90, 95, 120, 16, 94, 7.4], 148: [37, 90, 86, 120, 14, 97, 7.1], 149: [38, 80, 95, 121, 12, 99, 7.4], 150: [38, 93, 93, 120, 12, 96, 7.2], 151: [36, 65, 75, 120, 17, 96, 7.5], 152: [38, 85, 69, 120, 17, 99, 7.2], 153: [39, 61, 98, 120, 11, 98, 7.4], 154: [38, 87, 61, 121, 15, 98, 7.5], 155: [37, 98, 71, 120, 13, 96, 7.6], 156: [36, 65, 86, 121, 12, 94, 7.2], 157: [38, 77, 57, 120, 17, 99, 7.3], 158: [36, 80, 97, 121, 12, 100, 7.3], 159: [37, 92, 78, 120, 16, 93, 7.2], 160: [36, 96, 72, 121, 14, 96, 7.5], 161: [36, 74, 70, 120, 11, 93, 7.4], 162: [39, 60, 76, 121, 15, 93, 7.1], 163: [36, 79, 59, 121, 14, 97, 7.6], 164: [36, 83, 64, 120, 14, 97, 7.3], 165: [39, 73, 61, 121, 14, 93, 7.4], 166: [38, 61, 97, 120, 13, 100, 7.3], 167: [38, 95, 91, 121, 12, 100, 7.5], 168: [37, 98, 81, 120, 14, 94, 7.3], 169: [36, 81, 80, 120, 11, 96, 7.6], 170: [39, 85, 61, 121, 14, 93, 7.5], 171: [39, 55, 100, 121, 16, 99, 7.1], 172: [39, 78, 55, 120, 14, 98, 7.2], 173: [39, 100, 89, 120, 13, 100, 7.2], 174: [38, 98, 84, 121, 12, 95, 7.3], 175: [38, 90, 84, 121, 12, 94, 7.4], 176: [37, 90, 95, 120, 16, 94, 7.4], 177: [37, 90, 86, 120, 14, 97, 7.1], 178: [38, 80, 95, 121, 12, 99, 7.4], 179: [38, 93, 93, 120, 12, 96, 7.2], 180: [36, 65, 75, 120, 17, 96, 7.5], 181: [38, 85, 69, 120, 17, 99, 7.2], 182: [39, 61, 98, 120, 11, 98, 7.4], 183: [38, 87, 61, 121, 15, 98, 7.5], 184: [37, 98, 71, 120, 13, 96, 7.6], 185: [36, 65, 86, 121, 12, 94, 7.2], 186: [38, 77, 57, 120, 17, 99, 7.3], 187: [36, 80, 97, 121, 12, 100, 7.3], 188: [37, 92, 78, 120, 16, 93, 7.2], 189: [36, 96, 72, 121, 14, 96, 7.5], 190: [36, 74, 70, 120, 11, 93, 7.4], 191: [39, 60, 76, 121, 15, 93, 7.1], 192: [36, 79, 59, 121, 14, 97, 7.6], 193: [36, 83, 64, 120, 14, 97, 7.3], 194: [39, 73, 61, 121, 14, 93, 7.4], 195: [38, 61, 97, 120, 13, 100, 7.3], 196: [38, 95, 91, 121, 12, 100, 7.5], 197: [37, 98, 81, 120, 14, 94, 7.3], 198: [36, 81, 80, 120, 11, 96, 7.6], 199: [39, 85, 61, 121, 14, 93, 7.5], 200: [39, 55, 100, 121, 16, 99, 7.1], 201: [39, 78, 55, 120, 14, 98, 7.2], 202: [39, 100, 89, 120, 13, 100, 7.2], 203: [38, 98, 84, 121, 12, 95, 7.3], 204: [38, 90, 84, 121, 12, 94, 7.4], 205: [37, 90, 95, 120, 16, 94, 7.4], 206: [37, 90, 86, 120, 14, 97, 7.1], 207: [38, 80, 95, 121, 12, 99, 7.4], 208: [38, 93, 93, 120, 12, 96, 7.2], 209: [36, 65, 75, 120, 17, 96, 7.5], 210: [38, 85, 69, 120, 17, 99, 7.2], 211: [39, 61, 98, 120, 11, 98, 7.4], 212: [38, 87, 61, 121, 15, 98, 7.5], 213: [37, 98, 71, 120, 13, 96, 7.6], 214: [36, 65, 86, 121, 12, 94, 7.2], 215: [38, 77, 57, 120, 17, 99, 7.3], 216: [36, 80, 97, 121, 12, 100, 7.3], 217: [37, 92, 78, 120, 16, 93, 7.2], 218: [36, 96, 72, 121, 14, 96, 7.5], 219: [36, 74, 70, 120, 11, 93, 7.4], 220: [39, 60, 76, 121, 15, 93, 7.1], 221: [36, 79, 59, 121, 14, 97, 7.6], 222: [36, 83, 64, 120, 14, 97, 7.3], 223: [39, 73, 61, 121, 14, 93, 7.4], 224: [38, 61, 97, 120, 13, 100, 7.3], 225: [38, 95, 91, 121, 12, 100, 7.5], 226: [37, 98, 81, 120, 14, 94, 7.3], 227: [36, 81, 80, 120, 11, 96, 7.6], 228: [39, 85, 61, 121, 14, 93, 7.5], 229: [39, 55, 100, 121, 16, 99, 7.1], 230: [39, 78, 55, 120, 14, 98, 7.2], 231: [39, 100, 89, 120, 13, 100, 7.2], 232: [38, 98, 84, 121, 12, 95, 7.3], 233: [38, 90, 84, 121, 12, 94, 7.4], 234: [37, 90, 95, 120, 16, 94, 7.4], 235: [37, 90, 86, 120, 14, 97, 7.1], 236: [38, 80, 95, 121, 12, 99, 7.4], 237: [38, 93, 93, 120, 12, 96, 7.2], 238: [36, 65, 75, 120, 17, 96, 7.5], 239: [38, 85, 69, 120, 17, 99, 7.2], 240: [39, 61, 98, 120, 11, 98, 7.4], 241: [38, 87, 61, 121, 15, 98, 7.5], 242: [37, 98, 71, 120, 13, 96, 7.6], 243: [36, 65, 86, 121, 12, 94, 7.2], 244: [38, 77, 57, 120, 17, 99, 7.3], 245: [36, 80, 97, 121, 12, 100, 7.3], 246: [37, 92, 78, 120, 16, 93, 7.2], 247: [36, 96, 72, 121, 14, 96, 7.5], 248: [36, 74, 70, 120, 11, 93, 7.4], 249: [39, 60, 76, 121, 15, 93, 7.1], 250: [36, 79, 59, 121, 14, 97, 7.6], 251: [36, 83, 64, 120, 14, 97, 7.3], 252: [39, 73, 61, 121, 14, 93, 7.4], 253: [38, 61, 97, 120, 13, 100, 7.3], 254: [38, 95, 91, 121, 12, 100, 7.5], 255: [37, 98, 81, 120, 14, 94, 7.3], 256: [36, 81, 80, 120, 11, 96, 7.6], 257: [39, 85, 61, 121, 14, 93, 7.5], 258: [39, 55, 100, 121, 16, 99, 7.1], 259: [39, 78, 55, 120, 14, 98, 7.2], 260: [39, 100, 89, 120, 13, 100, 7.2], 261: [38, 98, 84, 121, 12, 95, 7.3], 262: [38, 90, 84, 121, 12, 94, 7.4], 263: [37, 90, 95, 120, 16, 94, 7.4], 264: [37, 90, 86, 120, 14, 97, 7.1], 265: [38, 80, 95, 121, 12, 99, 7.4], 266: [38, 93, 93, 120, 12, 96, 7.2], 267: [36, 65, 75, 120, 17, 96, 7.5], 268: [38, 85, 69, 120, 17, 99, 7.2], 269: [39, 61, 98, 120, 11, 98, 7.4], 270: [38, 87, 61, 121, 15, 98, 7.5], 271: [37, 98, 71, 120, 13, 96, 7.6], 272: [36, 65, 86, 121, 12, 94, 7.2], 273: [38, 77, 57, 120, 17, 99, 7.3], 274: [36, 80, 97, 121, 12, 100, 7.3], 275: [37, 92, 78, 120, 16, 93, 7.2], 276: [36, 96, 72, 121, 14, 96, 7.5], 277: [36, 74, 70, 120, 11, 93, 7.4], 278: [39, 60, 76, 121, 15, 93, 7.1], 279: [36, 79, 59, 121, 14, 97, 7.6], 280: [36, 83, 64, 120, 14, 97, 7.3], 281: [39, 73, 61, 121, 14, 93, 7.4], 282: [38, 61, 97, 120, 13, 100, 7.3], 283: [38, 95, 91, 121, 12, 100, 7.5], 284: [37, 98, 81, 120, 14, 94, 7.3], 285: [36, 81, 80, 120, 11, 96, 7.6], 286: [39, 85, 61, 121, 14, 93, 7.5], 287: [39, 55, 100, 121, 16, 99, 7.1], 288: [39, 78, 55, 120, 14, 98, 7.2], 289: [39, 100, 89, 120, 13, 100, 7.2], 290: [38, 98, 84, 121, 12, 95, 7.3], 291: [38, 90, 84, 121, 12, 94, 7.4], 292: [37, 90, 95, 120, 16, 94, 7.4], 293: [37, 90, 86, 120, 14, 97, 7.1], 294: [38, 80, 95, 121, 12, 99, 7.4], 295: [38, 93, 93, 120, 12, 96, 7.2], 296: [36, 65, 75, 120, 17, 96, 7.5], 297: [38, 85, 69, 120, 17, 99, 7.2], 298: [39, 61, 98, 120, 11, 98, 7.4], 299: [38, 87, 61, 121, 15, 98, 7.5], 300: [37, 98, 71, 120, 13, 96, 7.6], 301: [36, 65, 86, 121, 12, 94, 7.2], 302: [38, 77, 57, 120, 17, 99, 7.3], 303: [36, 80, 97, 121, 12, 100, 7.3], 304: [37, 92, 78, 120, 16, 93, 7.2], 305: [36, 96, 72, 121, 14, 96, 7.5], 306: [36, 74, 70, 120, 11, 93, 7.4], 307: [39, 60, 76, 121, 15, 93, 7.1], 308: [36, 79, 59, 121, 14, 97, 7.6], 309: [36, 83, 64, 120, 14, 97, 7.3], 310: [39, 73, 61, 121, 14, 93, 7.4], 311: [38, 61, 97, 120, 13, 100, 7.3], 312: [38, 95, 91, 121, 12, 100, 7.5], 313: [37, 98, 81, 120, 14, 94, 7.3], 314: [36, 81, 80, 120, 11, 96, 7.6], 315: [39, 85, 61, 121, 14, 93, 7.5], 316: [39, 55, 100, 121, 16, 99, 7.1], 317: [39, 78, 55, 120, 14, 98, 7.2], 318: [39, 100, 89, 120, 13, 100, 7.2], 319: [38, 98, 84, 121, 12, 95, 7.3], 320: [38, 90, 84, 121, 12, 94, 7.4], 321: [37, 90, 95, 120, 16, 94, 7.4], 322: [37, 90, 86, 120, 14, 97, 7.1], 323: [38, 80, 95, 121, 12, 99, 7.4], 324: [38, 93, 93, 120, 12, 96, 7.2], 325: [36, 65, 75, 120, 17, 96, 7.5], 326: [38, 85, 69, 120, 17, 99, 7.2], 327: [39, 61, 98, 120, 11, 98, 7.4], 328: [38, 87, 61, 121, 15, 98, 7.5], 329: [37, 98, 71, 120, 13, 96, 7.6], 330: [36, 65, 86, 121, 12, 94, 7.2], 331: [38, 77, 57, 120, 17, 99, 7.3], 332: [36, 80, 97, 121, 12, 100, 7.3], 333: [37, 92, 78, 120, 16, 93, 7.2], 334: [36, 96, 72, 121, 14, 96, 7.5], 335: [36, 74, 70, 120, 11, 93, 7.4], 336: [39, 60, 76, 121, 15, 93, 7.1], 337: [36, 79, 59, 121, 14, 97, 7.6], 338: [36, 83, 64, 120, 14, 97, 7.3], 339: [39, 73, 61, 121, 14, 93, 7.4], 340: [38, 61, 97, 120, 13, 100, 7.3], 341: [38, 95, 91, 121, 12, 100, 7.5], 342: [37, 98, 81, 120, 14, 94, 7.3], 343: [36, 81, 80, 120, 11, 96, 7.6], 344: [39, 85, 61, 121, 14, 93, 7.5], 345: [39, 55, 100, 121, 16, 99, 7.1], 346: [39, 78, 55, 120, 14, 98, 7.2], 347: [39, 100, 89, 120, 13, 100, 7

From the scatter plot generated below of function (2a), it is evident that there is no correlation between time stamps and pulse rate chosen randomly. Very few values (9 red values) are abnormal out of the 50 selected. Again, emphasizing a non-deterministic approach for selection of records.



As seen in the histogram for function 'frequencyAnalytics' and frequencyAnalytics2' (2b), there appears to be a random but slight normal distribution where majority of the pulse rates is between 60 and 99. The most common pulse rates in that sample seem to be between 75 and 80. There was only 1 record with pulse rate of 100. A total of 9 abnormal records and 41 normal records.

A random function is used to generate random numbers, within range of the number of samples generated in phase 1, to choose n number of time stamps to use as records, much like the previous function in 2a.

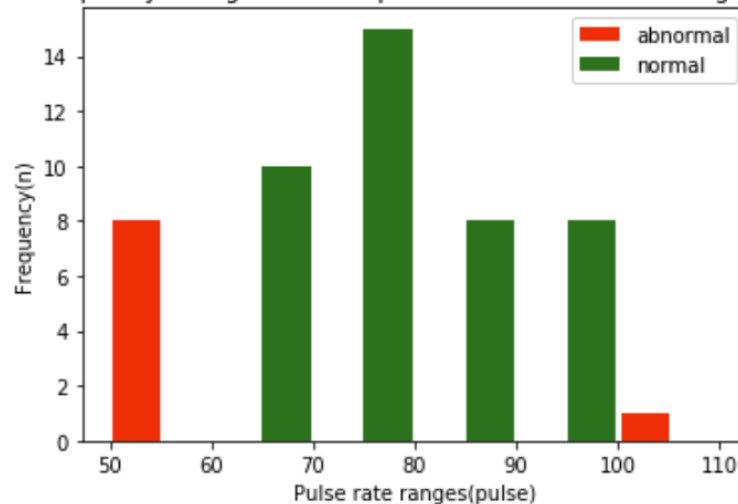
Space complexity is minimal as most of the data printed is stored in a dictionary. Many small lists are also used within the functions but there is the plotting and preparation ('for' loop) of the data for the histogram plot.

**2b function output:**

```
In [89]: frequencydict = frequencyAnalytics(mydict,50)
         print(frequencydict)
```

There are 50 samples selected.

Pulse frequency histogram of sample chosen from abnormalSignAnalytics.



Time taken to run function: 0.31235694885253906 seconds

{100: 1, 72: 1, 76: 1, 95: 1, 57: 3, 82: 2, 64: 2, 94: 1, 63: 2, 59: 3, 55: 2, 61: 2, 77: 3, 73: 2, 71: 1, 65: 1, 62: 1, 74: 2, 67: 1, 70: 3, 80: 1, 78: 1, 81: 2, 99: 2, 98: 3, 86: 1, 79: 1, 89: 1, 69: 1, 97: 1, 84: 1}

However, when using ‘frequencyAnalytics2’, where the code to plot the histogram was taken out and inputted into a new function ‘frequencyAnalyticsHisto’, the time to run the algorithm greatly improves from 0.321 seconds to 0.00087 seconds. I believe this is due to the time saved not printing and not having to use the additional ‘for loop’ to separate pulse rates into normal and abnormal lists before plotting.

```
In [38]: frequencydict2 = frequencyAnalytics2(mydict,100)
         print(frequencydict2)
```

There are 100 samples selected.

Time taken to run function: 0.0008740425109863281 seconds

{100: 2, 72: 3, 76: 2, 95: 4, 57: 5, 82: 3, 64: 2, 94: 1, 63: 3, 59: 3, 55: 3, 61: 3, 77: 4, 73: 2, 71: 2, 65: 3, 62: 1, 74: 2, 67: 1, 70: 6, 80: 1, 78: 1, 81: 2, 99: 3, 98: 4, 86: 5, 79: 1, 89: 1, 69: 1, 97: 1, 84: 1, 75: 3, 92: 3, 66: 1, 85: 2, 83: 3, 56: 2, 93: 4, 60: 1, 68: 2, 96: 1, 87: 1, 90: 1}

**3a function output:**

```
In [111]: healthanalyser = healthAnalyzer(mydict, 56)
          print(healthanalyser)
```

```
Time taken to run function: 0.0007848739624023438 seconds
[(1, 39, 61, 56, 121, 11, 93, 7.2), (75, 36, 83, 56, 120, 12, 99, 7.2), (126, 37, 90, 56, 121, 12, 98, 7.2), (202, 3
9, 80, 56, 121, 14, 99, 7.3), (240, 39, 92, 56, 121, 14, 100, 7.4), (310, 37, 84, 56, 121, 13, 96, 7.5), (337, 36, 7
0, 56, 121, 12, 93, 7.1), (457, 37, 100, 56, 121, 17, 99, 7.3), (514, 39, 76, 56, 121, 11, 93, 7.5), (548, 36, 79, 5
6, 121, 13, 95, 7.2), (551, 37, 98, 56, 120, 16, 96, 7.5), (575, 36, 78, 56, 121, 15, 93, 7.6), (597, 39, 77, 56, 12
1, 15, 94, 7.4), (626, 38, 86, 56, 121, 15, 98, 7.1), (673, 38, 76, 56, 121, 12, 95, 7.2), (700, 38, 89, 56, 120, 12,
95, 7.2), (740, 38, 83, 56, 120, 15, 97, 7.3), (756, 37, 84, 56, 120, 16, 98, 7.4), (786, 39, 70, 56, 120, 15, 93, 7.
4), (808, 37, 82, 56, 121, 17, 98, 7.4), (929, 38, 76, 56, 120, 16, 95, 7.2), (964, 38, 82, 56, 121, 14, 96, 7.4)]
```

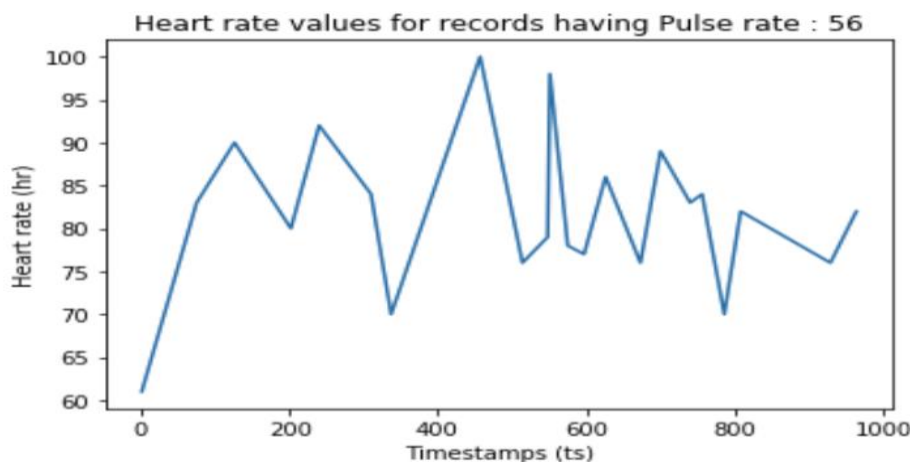
**3b)** The 'sorted()' function used in python, I believe is defaulted to the Timsort function so time complexity is worst-case  $O(n \log n)$  and best-case is  $O(n)$ , space complexity is  $O(n)$ .

I also use a linear search algorithm which has a worst-case  $O(n)$  time complexity.

Overall, worst-case scenario is  $O(n \log n)$  due to the Timsort function used.

However, if the outputted multidimensional list is not required to be in a sorted order then 'healthAnalyser' would only use a linear search which would have a lower complexity of  $O(n)$ .

Below is the plotted heart rate values for records having pulse rate 56 which shows little correlation with timestamps.



Another solution, 'healthAnalyzer2', utilises a binary search in addition to the Timsort used in 'healthAnalyzer1'. This has a worst-case scenario of  $O(\log n)$  which is better than Timsort. It also has a better  $O(1)$  space complexity. However, this algorithm's complexity is decided on the sorting of the list which utilises a 'for loop' so the complexity is  $O(n)$  overall.

**4a function output:**

```
print(benchmarking())
#nvalues = [1000,2500,5000, 7500, 10000]
```

```
[0.01545095443725586, 0.039411306381225586, 0.09513401985168457, 0.09292006492614746, 0.13114500045776367]
```

4b) Above are the benchmarks taken for 'myHealthcare' and the times are shown in seconds. There shows a positive correlation between time and 'n' number of operations. This function has a  $O(n)$  time complexity as there are only 'for loops' used to generate the samples. Data generated is stored in a dictionary.

Below, is the plot of the above results. It generally shows a steep linear increase as operations increase. There is however a small decline between 5000 and 7500 operations.

