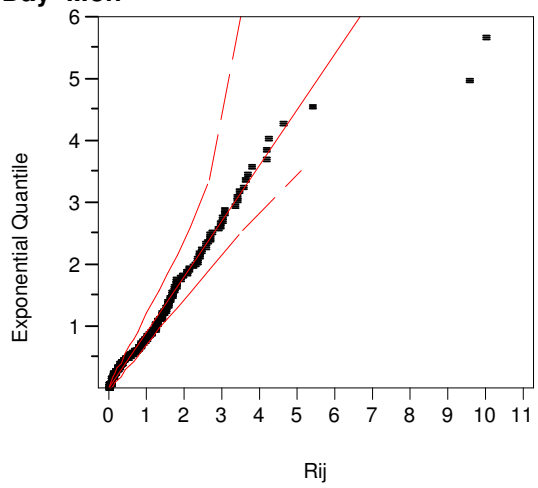
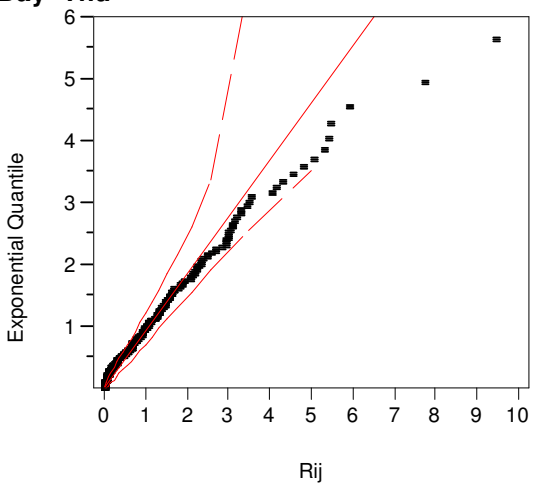
Appendix A-1 - Exponential Quantile Plots for Test 1

Quantile Plot: 0700-0703 for days in June

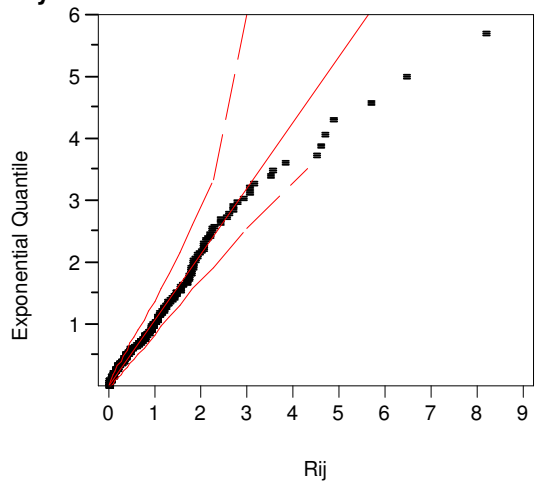
Day=Mon



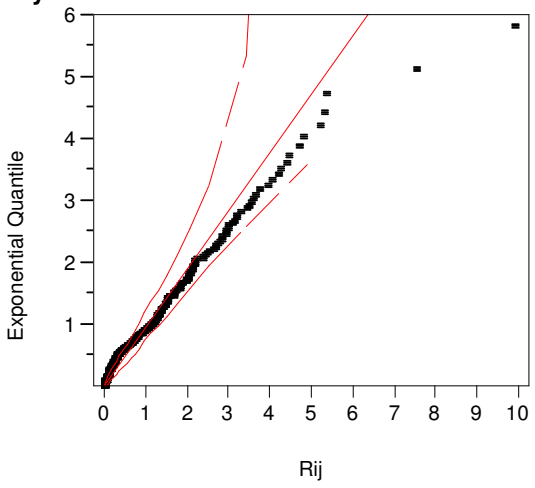
Day=Thu



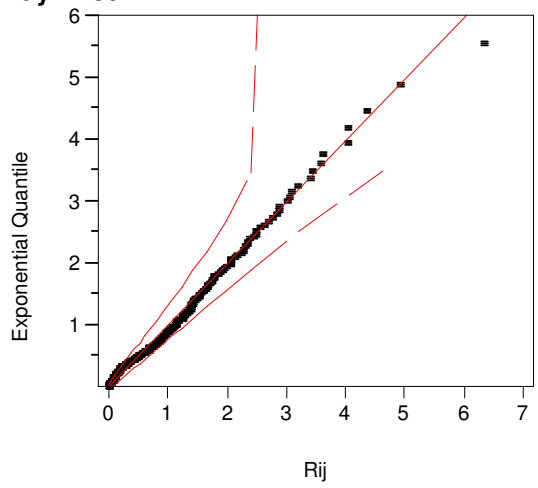
Day=Tue



Day=Fri

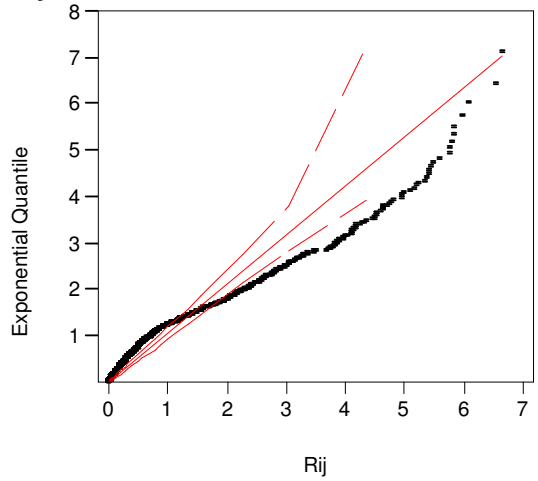


Day=Wed

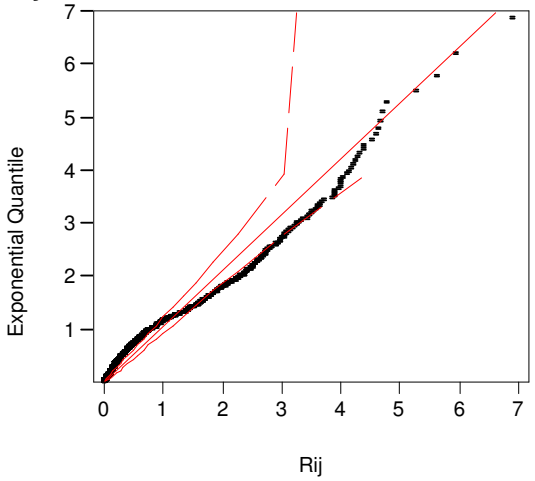


Quantile Plot: 1000-1003 for days in June

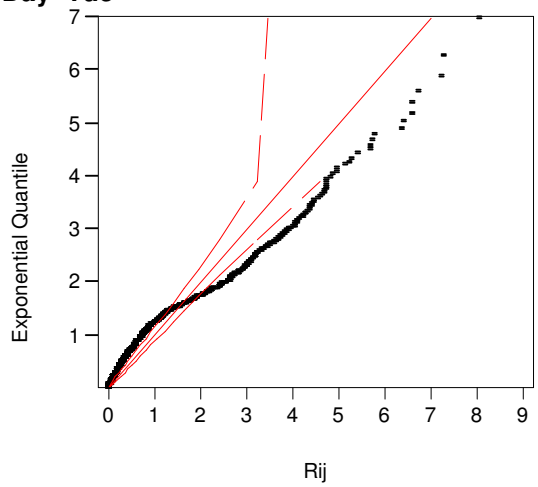
Day=Mon



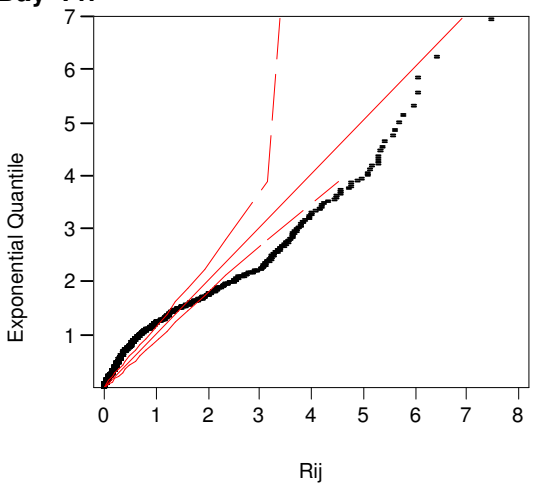
Day=Thu



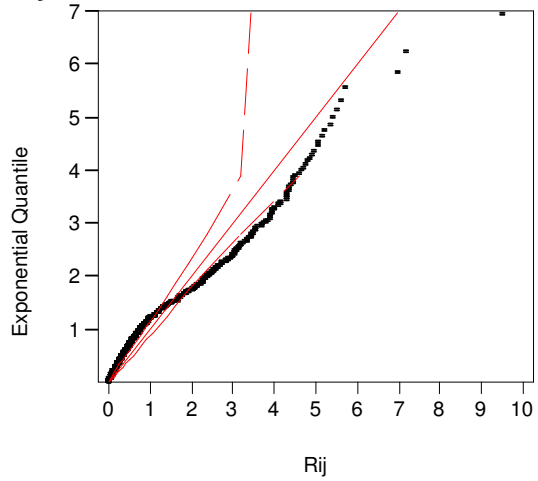
Day=Tue



Day=Fri

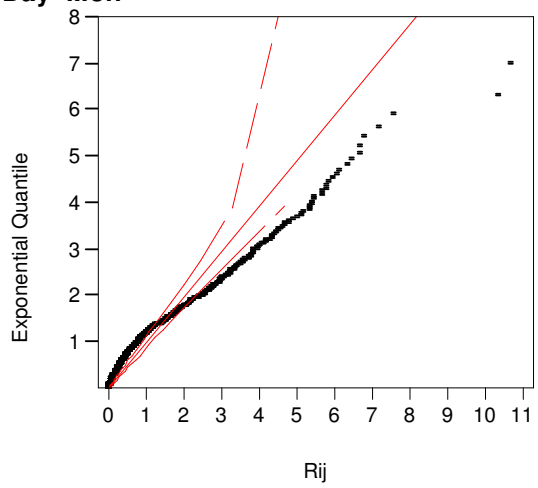


Day=Wed

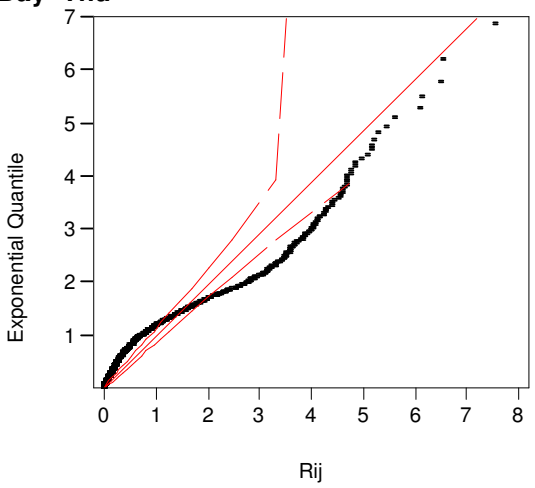


Quantile Plot: 1200-1203 for days in June

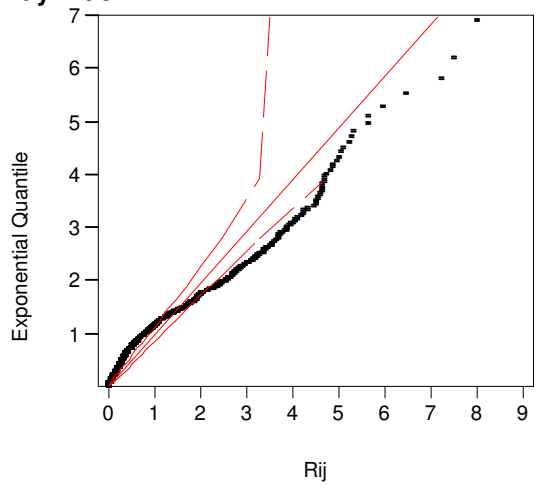
Day=Mon



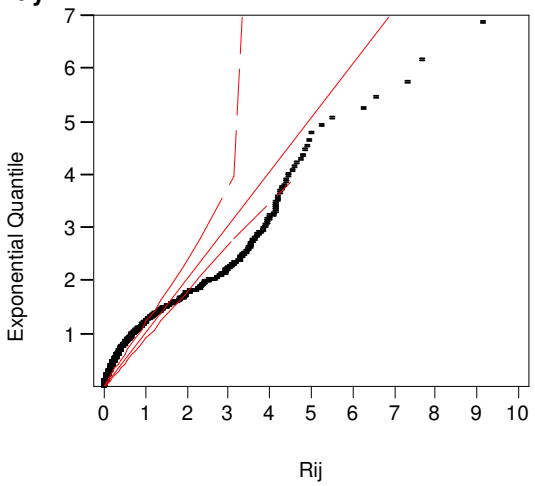
Day=Thu



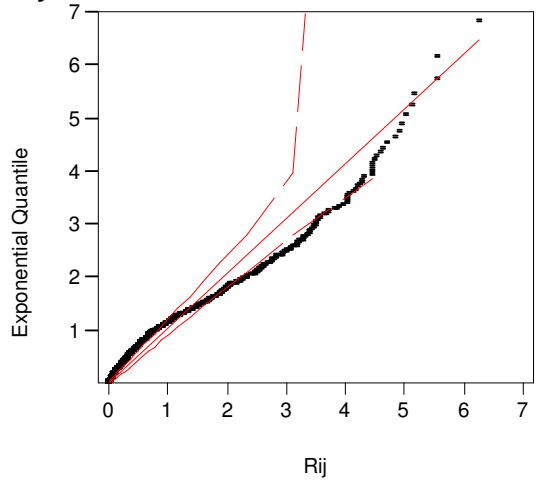
Day=Tue



Day=Fri

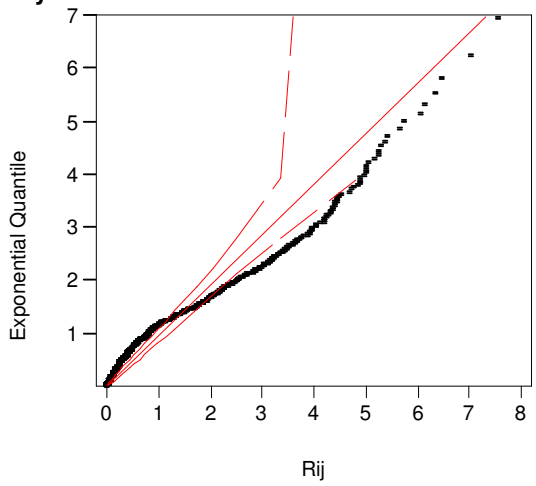


Day=Wed

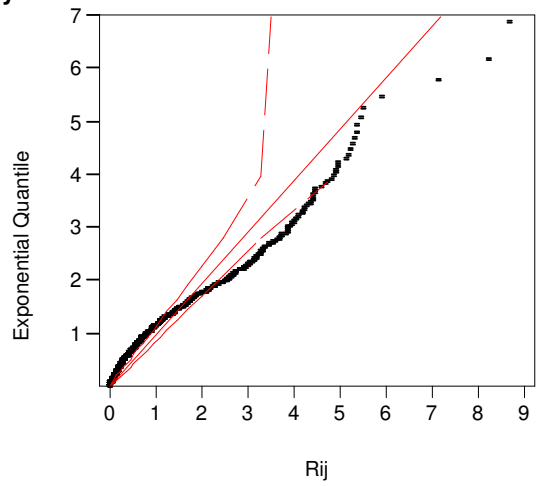


Quantile Plot: 1500-1503 for days in June

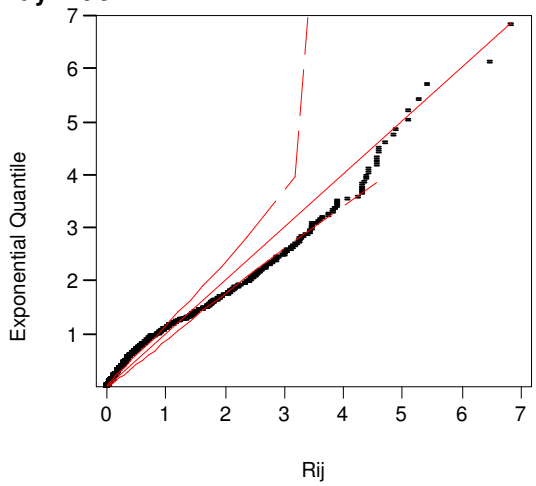
Day=Mon



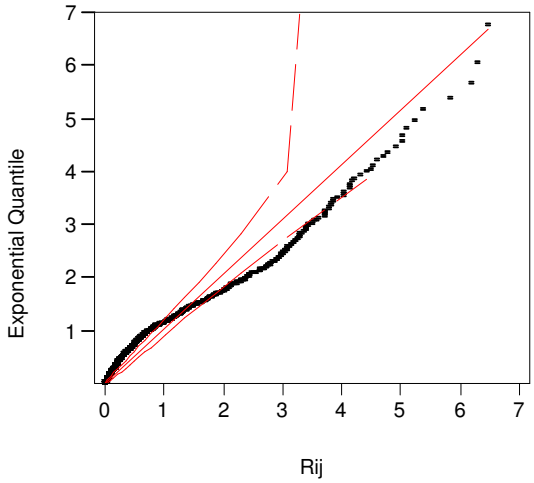
Day=Thu



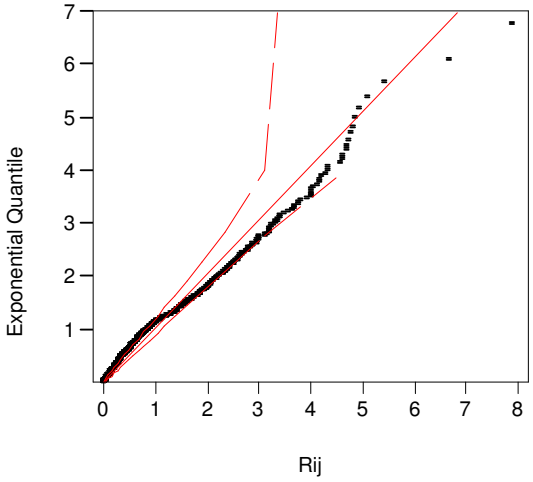
Day=Tue



Day=Fri
Quantile Plot

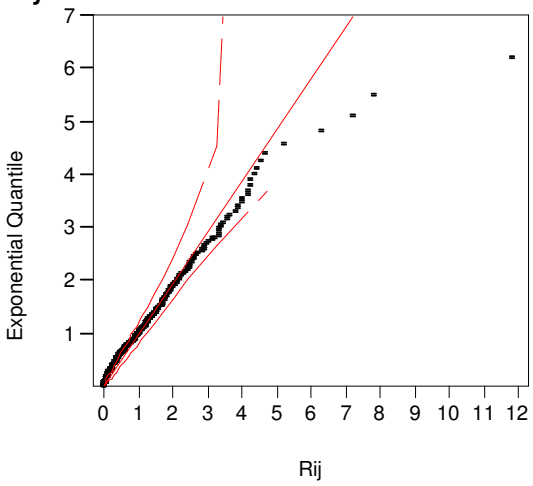


Day=Wed
Quantile Plot

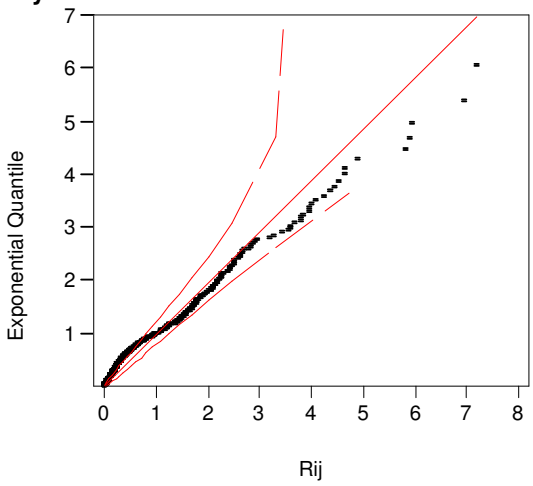


Quantile Plot: 1800-1803 for days in June

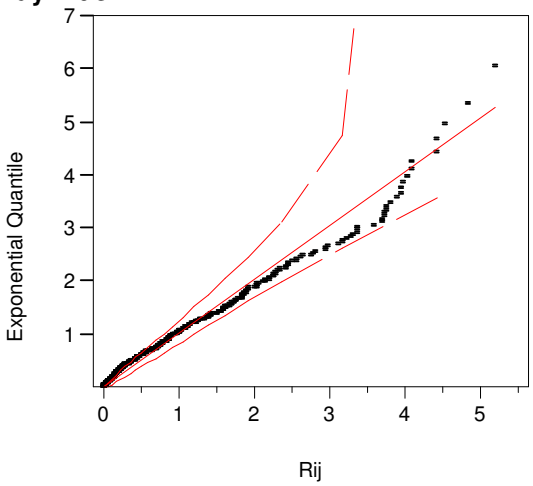
Day=Mon



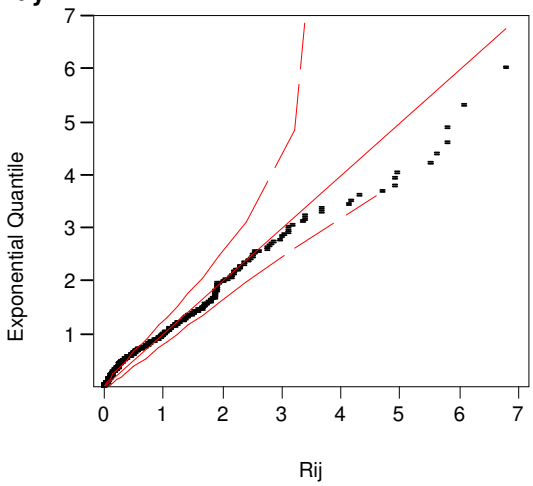
Day=Thu



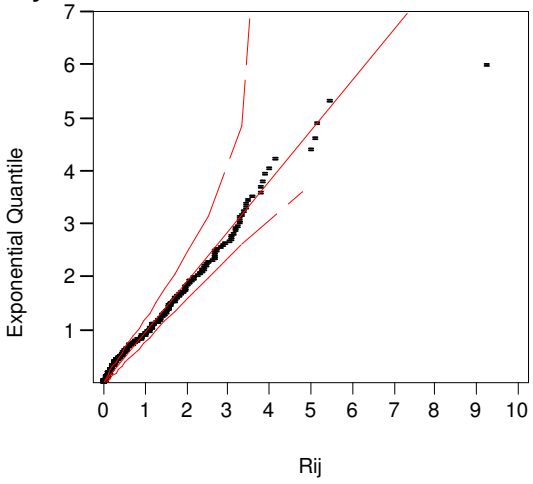
Day=Tue



Day=Fri

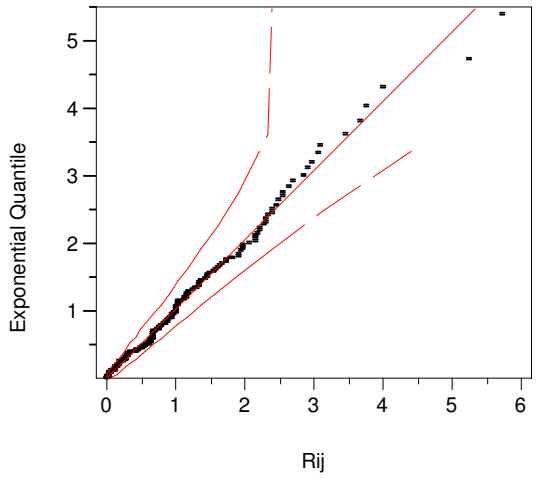


Day=Wed

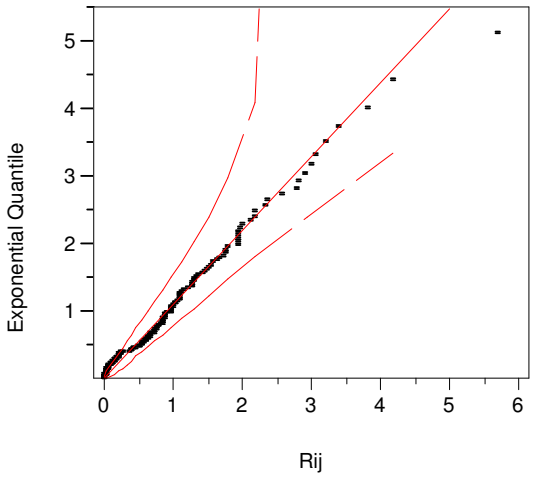


Quantile Plot: 2200-2203 for days in June

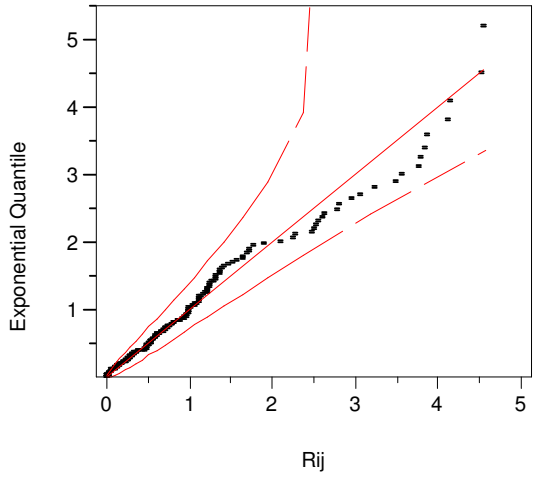
Day=Mon



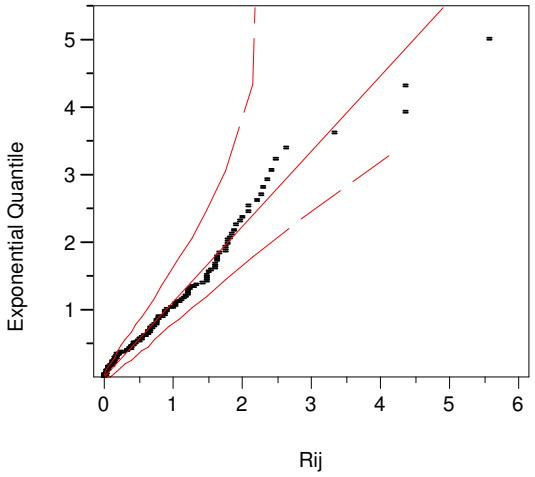
Day=Thu



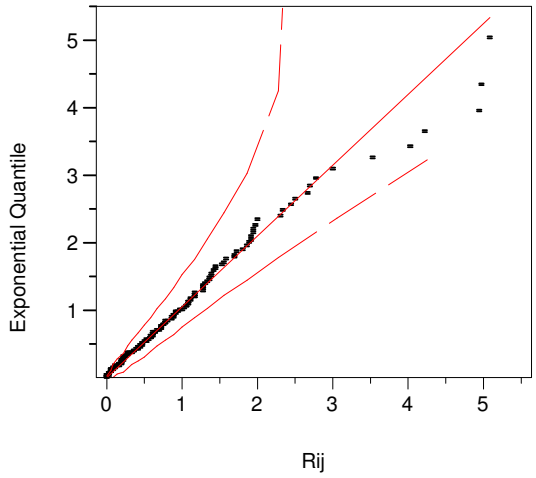
Day=Tue



Day=Fri



Day=Wed



Appendix A-2 - Macros to Calculate Chi-Square Statistics for Test 2

```
Sub Chisquare()
'
' Setup Macro
' Macro recorded 2/6/2004 by Yen Chu Cheng
'
'

    Const numBinForCalcPurposes = 49
    Const sizeBins = 1
    Const testDuration = 360
    Const startTime = 25200
    Const numBins = 5
    Const numIntervals = testDuration / sizeBins

    Dim currentDate
    Dim nextDate
    Dim currentTime
    Dim nextTime

    Dim numCalls
    Dim numSecWithCalls
    Dim numSecWithManyCalls
    Dim numInBin(0 To numBinForCalcPurposes)

    Dim offset
    Dim i
    Dim numberRows
    Cells(1, 1).Select
    Selection.End(xlDown).Select
    numberRows = ActiveCell.Row
    Selection.End(xlUp).Select

    'Sort first two rows
    ActiveCell.offset(0, 0).Columns("A:B").EntireColumn.Select
    Selection.Sort Key1:=ActiveCell, Order1:=xlAscending, Key2:=ActiveCell. _
        offset(0, 1).Range("A1"), Order2:=xlAscending, Header:=xlGuess,
OrderCustom _
        :=1, MatchCase:=False, Orientation:=xlTopToBottom, DataOption1:= _
        xlSortNormal, DataOption2:=xlSortNormal

    'Change second column from hh:mm:ss to seconds
    ActiveCell.offset(1, 2).Range("A1").Select
    ActiveCell.FormulaR1C1 = "=QUOTIENT(RC[-1],10000)*3600 +
QUOTIENT(MOD(RC[-1],10000),100)*60 + MOD(RC[-1],100)"

    'Copy to the rest of the rows
    ActiveCell.Select
    Selection.Copy
    ActiveCell.offset(0, -1).Range("A1").Select
    Selection.End(xlDown).Select
    ActiveCell.offset(0, 1).Range("A1").Select
    Range(Selection, Selection.End(xlUp)).Select
    ActiveSheet.Paste

    'Copy and paste special of numbering to remove dependency on each other
    Range("C2").Select
    Range(Selection, Selection.End(xlDown)).Select
    Selection.Copy
    Range("D2").Select
```

```
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone,
SkipBlanks _
        :=False, Transpose:=False

    'Put in formula to number the calls for each second
    ActiveCell.offset(0, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "=IF(ROW()=2,1,IF(QUOTIENT(RC[-1]-" & startTime
& "," & sizeBins & ")<>QUOTIENT(R[-1]C[-1]-" & startTime & "," & sizeBins &
"),1,R[-1]C+1))"

    'Copy to the rest of the rows
    ActiveCell.Select
    Selection.Copy
    ActiveCell.offset(0, -1).Range("A1").Select
    Selection.End(xlDown).Select
    ActiveCell.offset(0, 1).Range("A1").Select
    Range(Selection, Selection.End(xlUp)).Select
    ActiveSheet.Paste

    'Copy and paste special of numbering to remove dependency on each other
    Range("E2").Select
    Range(Selection, Selection.End(xlDown)).Select
    Selection.Copy
    Range("F2").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone,
SkipBlanks _
        :=False, Transpose:=False

    'Delete useless columns
    Cells(1, 2).Select
    Selection.EntireColumn.Delete
    Cells(1, 2).Select
    Selection.EntireColumn.Delete
    Cells(1, 3).Select
    Selection.EntireColumn.Delete

    'Initializing the array and number of calls
    For i = 0 To numBinForCalcPurposes
    numInBin(i) = 0
    Next i
    numCalls = 0
    numSecWithCalls = 0
    numSecWithManyCalls = 0

    currentDate = Cells(2, 1).Value
    currentTime = Cells(2, 2).Value
    For i = 2 To numberRows

        If Cells(i, 3).Value < Cells(i + 1, 3).Value Then
            Cells(i, 3).Select
            Selection.EntireRow.Delete
            i = i - 1

        Else
            numInBin(Cells(i, 3).Value) = numInBin(Cells(i, 3).Value) + 1

            nextDate = Cells(i + 1, 1).Value
            nextTime = Cells(i + 1, 2).Value
            If (nextDate <> currentDate) Or (((nextTime - startTime) \
testDuration) <> ((currentTime - startTime) \ testDuration)) Then
```

```
                For j = 0 To numBinForCalcPurposes
                    numCalls = j * numInBin(j) + numCalls
                    numSecWithCalls = numSecWithCalls + numInBin(j)
                    If j >= (numBins - 1) Then
                        numSecWithManyCalls = numSecWithManyCalls +
numInBin(j)
                    End If
                Next j
                numInBin(0) = numIntervals - numSecWithCalls

                'Print first column
                Cells(1 + offset * (numBins + 3), 5).Formula = "Arrivals"
                For j = 1 To (numBins - 1)
                    Cells(j + 1 + offset * (numBins + 3), 5).Formula = j - 1
                Next j
                Cells(numBins + 1 + offset * (numBins + 3), 5).Formula = ">"
& numBins - 2
                Cells(numBins + 2 + offset * (numBins + 3), 5).Formula =
"Lambda"


                'Print second column
                Cells(1 + offset * (numBins + 3), 6).Formula = "Actual Freq"
                For j = 1 To (numBins - 1)
                    Cells(j + 1 + offset * (numBins + 3), 6).Formula =
numInBin(j - 1)
                Next j
                Cells(numBins + 1 + offset * (numBins + 3), 6).Formula =
numSecWithManyCalls
                Cells(numBins + 2 + offset * (numBins + 3), 6).Formula =
numCalls / numIntervals

                'Print third column
                Cells(1 + offset * (numBins + 3), 7).Formula = "Probability"
                For j = 1 To (numBins - 1)
                    Cells(j + 1 + offset * (numBins + 3), 7).Formula =
"=EXP(-R[" & (numBins + 1 - j) & "]C[-1])*(R[" & (numBins + 1 - j) & "]C[-
1])^RC[-2]/FACT(RC[-2])"
                Next j
                Cells(numBins + 1 + offset * (numBins + 3), 7).Formula = "=1-
SUM(R[-" & (numBins - 1) & "]C:R[-1]C)"

                'Print fourth column
                Cells(1 + offset * (numBins + 3), 8).Formula = "Theoretical"
                Cells(2 + offset * (numBins + 3), 8).Range("A1:A" &
numBins).Formula = "=" & numIntervals & "*RC[-1]"

                'Print fifth column
                Cells(1 + offset * (numBins + 3), 9).Formula = "(Actual-
Theoretical)^2/Theoretical"
                Cells(2 + offset * (numBins + 3), 9).Range("A1:A" &
numBins).Formula = "=(RC[-3]-RC[-1])^2/RC[-1]"
                Cells(numBins + 2 + offset * (numBins + 3), 9).Formula =
"=SUM(R[-" & numBins & "]C:R[-1]C)"

                'Rearrange required values nicely
                Cells(offset + 1, 13) = Cells(numBins + 2 + offset * (numBins
+ 3), 9).Value

                'Initializing the array and number of calls and put in the
first value
                For j = 0 To numBinForCalcPurposes
```

```
                    numInBin(j) = 0
                Next j
                numCalls = 0
                numSecWithCalls = 0
                numSecWithManyCalls = 0

                'Increase offset
                offset = offset + 1

                currentDate = nextDate
                currentTime = nextTime

            End If
        End If
    Next i



End Sub
```