

21 : 既存の知識・技能の集成として・

2016年09月 加藤 厚 (2016年12月改訂)

21-1 タッチ選択した2つのカテゴリ変数・

のクロス集計結果を新頁に出力し、その値以上の「連関」が偶然のみで発生する確率を関数で求める script を作成します。

21-2 Script としては・

以下の通りの「既出@No.+α」:

- ①外部ファイル配列のタッチ選択は既出@19 checkbox 参照のための<form>は+α※ ※19では<form>なし (vs. 09では<form>あり)。
- ②2次元配列での計数は既出@19 上限を mxc と mxr とする点は+α※ ※09では0~10で「上限固定」。
- ③カイ2乗値と連関係数は既出@09 Yates の修正※、最小期待度数などは+α ※2項分布@11=離散分布を正規分布@14=連続分布で適切に近似するための修正@2×2集計表。「期待度数が5以下のセルが全体の20%以上 or 最小期待度数が1以下のセルが1つ以上ある場合は修正カイ2乗のpを用いるべき」という意見がある。
- ④確率関数の利用は既出@15 (t分布) カイ2乗関数の利用は+α※ ※正規分布からカイ2乗分布を導出。

21-3 資料49を実行すると資料50の・

- ①、性別×世代で②の結果が示されます※。
- ※principle.jsも既出+α・・「世代×主義」は09で既出の架空 data ですが、選択用に「性別」(架空)を追加

資料50 資料49の起動時及び選択・実行後の状態

「偶然発生する確率」は数表※でも得られますが、関数で計算できればより便利です。  
 ※自由度2の場合の確率と基準値は5%が5.99、2.5%が7.38、1%が9.21、0.5%が10.6です。

関連情報：超幾何関数を用いた確率分布の計算  
[http://www.geocities.jp/ikuro\\_kotaro/koramu/tyokika.htm](http://www.geocities.jp/ikuro_kotaro/koramu/tyokika.htm)

```

<title>カイ2乗関数</title><!-- saved from url=(0008)about:internet -->
<script src=principle.js></script>
<form 上が表頭・下が表側の変数:<input type="button" value="OK" onClick="表0">
<input type="checkbox" >行%</form>
<script>
列=変数の数, 件=dataの件数 (-1は変数名の1行)
列=v[0].length; 件=v.length-1; 桁=10000; document.write("<form><select>")
for (i=0; i<=列-1; i++) {document.write("<option>" +v[0][i])}
document.write("</select></form>"); document.write("<form><select>")
for (i=0; i<=列-1; i++) {document.write("<option>" +v[0][i])}
document.write("</select></form>")
function 表0 {
  頭=document.forms[1].elements[0].selectedIndex
  側=document.forms[2].elements[0].selectedIndex
  新=window.open("", "", "width=720, height=360, scrollbars=yes, resizable=yes")
  t0="<title>クロス集計表: "+v[0][頭]+"/x "+v[0][側]+"/</title><pre>表頭: No. "
  t0=t0+(頭+1)+" "+v[0][頭]+"/n 表側: No. "+(側+1)+" "+v[0][側]+"/n">
  欠=0; mxc=0; mxr=0; 新.moveBy(300, 200); 新.document.write(t0)
  for (i=1; i<=件; i++) {if (null==v[i][頭] || null==v[i][側]) continue
  if (mxc<v[i][頭] mxc=v[i][頭]; if (mxr<v[i][側] mxr=v[i][側])
  }
  cc=new Array 0; rt=new Array 0; ct=new Array 0
  for (i=0; i<=mxc; i++) cc[i]=new Array 0
  for (i=0; i<=mxc; i++) {for (j=0; j<=mxr; j++) {cc[i][j]=0}}
  for (c=0; c<=mxc; c++) ct[c]=0; for (r=0; r<=mxr; r++) rt[r]=0
  for (i=1; i<=件; i++) {
    x=v[i][頭]; y=v[i][側]; if (null==x || null==y) {欠++; continue}
    cc[x][y]++; ct[x]++; rt[y]++
  }
  有=件-欠; chi2=0; yc2=0; mnfe=999; le5fe=0; vcm=0; hp="(100%"
  新.document.write("<table border=1><tr align='center'><td>値</td>")
  for (c=0; c<=mxc; c++) {if (0==ct[c]) continue
  新.document.write("<td>","c,"</td>"); vcm++
  }
  新.document.write("<td>計</td></tr>"); vrw=0
  for (r=mxr; 0<=r; r--) {if (0==rt[r]) continue
  新.document.write("<tr align='center'><td>","r,"</td>"); vrw++
  for (c=0; c<=mxc; c++) {if (0==ct[c]) continue
  fe=rt[r]*ct[c]/有; if (fe<mnfe) mnfe=fe
  x=cc[c][r]-fe; yx=Math.abs(x); if (5<yx) {yx=yx-5} else yx=0
  chi2=chi2+x*x/fe; yc2=yc2+y*x/yx/fe; if (fe<5) le5fe++
  t1="<td>" +cc[c][r]; p=" "+Math.round(1000*cc[c][r]/rt[r])/10+"%"
  if (document.forms[0].elements[1].checked==true) t1=t1+p
  新.document.write(t1+"</td>")
  t2="<td>" +rt[r]; if (document.forms[0].elements[1].checked==true) t2=t2+hp
  新.document.write(t2+"</td></tr>")
  }
  新.document.write("<tr align='center'><td>計</td>"); pct="""
  for (c=0; c<=mxc; c++) {if (0==ct[c]) continue
  t3="<td>" +ct[c]; p=" "+Math.round(1000*ct[c]/有)/10+"%"
  if (document.forms[0].elements[1].checked==true) {t3=t3+p; pct=hp}
  新.document.write(t3+"</td>")
  }
  新.document.write("</td></tr></table><hr>")
  k=vrw; if (vcm<vrw) k=vcm; cr=Math.sqrt(chi2/((k-1)*有)); df=(vrw-1)*(vcm-1)
  r="欠測"-欠+ 連関係数(V)=" +Math.round(1000*cr)/1000+"%"
  pc=Math.round(桁*chi(chi2, df))/桁; py=Math.round(桁*chi(yc2, df))/桁
  r="カイ2乗 (" +df+ )" = "+Math.round(1000*chi2)/1000+" p=" +pc+"%"
  if (vrw=2&&vcm=2) r=r+"修正 " +Math.round(1000*yc2)/1000+" p=" +py+"%"
  mnfe=Math.round(1000*mnfe)/1000; l5p=Math.round(100000*le5fe/(vrw*vcm))/1000
  新.document.write(r+"最小期待度数=" +mnfe+" 期待度数≤5のセル: "+l5p+"%<n")
  }
function chi(値, df) {
  px=0; div=df/2; res=Math.floor(div)
  if (div=res) {
    if (df!=2) {for (i=df-2; 1<i; i-=2) {px=(px+1)*値/i}
    prb=Math.exp(-.5*値)*(px+1)}
  } else {rch=Math.sqrt(値); prb=正規(rch)
  if (3<df) {for (i=df-2; i>=3; i-=2) {px=(px+1)*値/i}
  if (2<df) {prb=Math.sqrt(2/Math.PI)*Math.exp(-.5*値)*(px+1)*rch*prb}
  return prb}
function 正規(z) {A= 0.00005383; B= 0.000488906; C= 0.000380036; D= 0.032776263
E= 0.211410061; F= 0.49867347; xz=(((z*A+B)*z+C)*z+D)*z+E)*z+F)*z+1
prb=Math.exp(-16*Math.log(xz)); if (prb>1) prb=0; return prb}
</script>

```

外部ファイルは同一 folder の principle.js

参照には要<form>

列=変数の数, 件=dataの件数 (-1は変数名の1行)

新(出力) 頁を動作指定頁に対し右300 下200の位置に移動

実 data に基づく最大値の判定

cc=cell count (区)、rt=row total (行計)、ct=column total (列計)。

vcm=valid※ column ※<0< 該当件数

最小期待度数: 各セルの期待度数を求め、事前の値より小さければ入れ替える。

Yatesの連続性修正

fe≤5のセルの数

□行%がチェックされていたら行%を各セルの出力に追加

正規分布からカイ2乗分布の確率を導出する近似式: この分野の数学に詳しくない人には理解不能。 :-)

資料49 カテゴリ変数間の連関の強さを確率関数で求める JavaScript の例