B

An Annotated MATHEMATICA Notebook for a Comparison of Approximations to the Exact Density of R

The following parts of code are annotated for better reproducibility of the results and potential adaptations where needed. The annotations are kept in roman font type and should not be confused with the actual code presented in `typewriter font`. The code reproduced is complete so that it can be transcribed to instantly work with MATHEMATICA Version 4 or later. The actual version used to produce the results reported in Section 7.5.2 was MATHEMATICA 4.0.1.0 on a windows platform but the code was also tested and works with Version 3.0. For better comprehension, the code is sectioned in a general part that comes first and then code pertaining to the single approximations.

General part. First, the degrees of freedom (as an example, 48 is used in the code) and the value of ρ (as an example, $\rho = .20$ is used) for the comparisons are fixed. Note that the degrees of freedom are $\mathrm{df} = n - 2$ so that for a situation with 50 persons, for example, a value of 48 has to be inserted. Also, in some of the functions, the degrees of freedom appear as ν (or in `typewriterfont` as `nu`). Of course, this symbol should not be confused with the standard error of effect size estimators as introduced in the text.

As another preliminary step, the standard package for continuous statistical distributions is loaded and the density of the noncentral t distribution is defined in the following code.

```
df = 48

ActualRho = .2

<< Statistics`ContinuousDistributions`

DensityStudentT[x_, nu_] := PDF[StudentTDistribution[nu], x]

DensityNoncentralT[x_, nu_, delta_] :=
          PDF[NoncentralStudentTDistribution[nu,delta],x]
```

Hotelling's exact density. First, the code to specify the density of r as given by Hotelling (1953) is presented (see also Equation 3.1).

```
TheoreticalRDensityHotelling[r_,nu_,rho_]:=
     nu/Sqrt[2 Pi]Beta[nu + 1, 1/2]/Gamma[1/2]
     (1 - rho^2)^(1/2(nu + 1))(1 - r^2)^(1/2  nu - 1)
     (1 - rho r)^(1/2 - (nu + 1))
     Hypergeometric2F1[1/2, 1/2,nu + 3/2, (1 + rho r)/2]
```

As can be seen, this is the exact density. It should be noted that the hypergeometric function is at some points numerically somewhat fragile, that is, it leads in the region of the singularity to unreliable values. For the present situations very high values for ρ (e.g., $\rho \geq .90$) in combination with large values for the degrees of freedom (e.g., df $\geq$ 250) may cause computational problems. Nevertheless, except for these borderline cases the specified function for the theoretical density given above works perfectly well. However, to avoid numerical problems the value of the hypergeometric function can be approximated to a very high and estimable degree. First, the approximation is computed as a truncation of the Taylor series expansion at the seventh term

```
Normal[Series[Hypergeometric2F1[a, b, c, x], {x, 0, 7}]]
```

The result is very large in expression and is subsequently defined as

```
Hgf[a_, b_, c_, x_] := 1 + (abx)/c +
                       (a(1 + a)b(1 + b)x^2)/(2 c (1 + c)) + ...
```

which is truncated as given, indicated by "..." The rest of the result from the step before has to be inserted instead of "...". One may now wish to estimate the error caused by this truncation. The error caused by truncating the series at any stage is less than $2/(1-\rho r)$ times the last term used (Hotelling, 1953, p. 200). For the proposed truncation the error can therefore be estimated by

```
LastTerm[a_, b_, c_, x_] := (a (1 + a) (2 + a) (3 + a) (4 + a) (5 + a)
   (6 + a) b (1 + b) (2 + b) (3 + b) (4 + b) (5 + b) (6 + b) x^7)/
   (5040 c (1 + c) (2 + c) (3 + c) (4 + c) (5 + c) (6 + c))

UpperBoundForErrorCausedByTruncation[r_, nu_, rho_] :=
          2/(1 - rho r) LastTerm[1/2, 1/2, nu + 3/2, (1 + rho r)/2]
```

In the present case this error is approximately 1.4151^{-11} for a value of $r = 1$ which is also the maximum of error. This can be easily seen by inspecting a plot of the error for varying r, which produces for the present case Figure B.1.

```
Plot[UpperBoundForErrorCausedByTruncation[x, df, ActualRho],
    {x, -1, 1}, PlotRange -> All, AxesOrigin -> {-1.01, 0}]
```

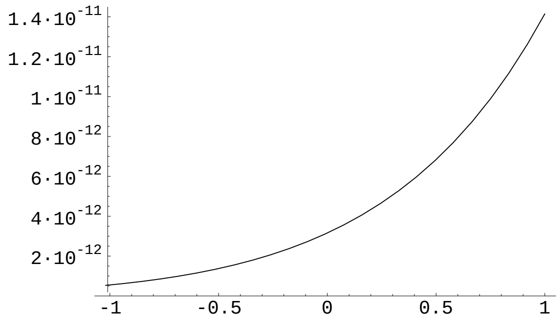

Figure B.1 Upper bounds of truncation error for the hypergeometric series used in the computation of the exact density for varying r, df$=48$, and $\rho = .20$.

The error is obviously very small for all values of r and has its maximum on the interval $[-1, 1]$ at 1 which is still very small in value. The truncation can therefore safely be used. The modified density of r can now be defined as

```
RDensityHotelling[r_, nu_, rho_] :=
      1/(Pi Sqrt[2])(1 - r^2)^((nu/2) - 1) nu
      (1 - r rho)^(-nu - (1/2)) (1 - rho^2)^(1/2(nu + 1))
      Beta[nu + 1, 1/2] Hgf[1/2, 1/2,nu + 3/2, (r rho + 1)/2]
```

where only the hypergeometric function is substituted by the truncated version. Using this form for the density of r the expected values and variances that are used as criteria values for all the following approximations are computed by

```
ExpectationOfHotellingsR[nu_, rho_] :=
  NIntegrate[x RDensityHotelling[x, nu, rho], {x, -0.99999, 0.99999}]

ExpectationOfHotellingsR[df, ActualRho]

SecondMomentOfHotellingsR[nu_, rho_] :=
  NIntegrate[x^2 RDensityHotelling[x, nu, rho],
  {x, -0.99999, 0.99999}]

VarOfHotellingsR[nu_, rho_] :=
       SecondMomentOfHotellingsR[nu, rho] -
       (ExpectationOfHotellingsR[nu, rho])^2

VarOfHotellingsR[df, ActualRho]
```

resulting in values of 0.198047 for the expected value and 0.0188894 for the variance. Again, the density can be plotted for inspection by

```
P2 = Plot[RDensityHotelling[x, df, ActualRho], {x, -1, 1},
     PlotRange -> All, AxesOrigin -> {0, 0},
     PlotStyle -> {RGBColor[0, 0, 0]}]
```

resulting in Figure B.2.

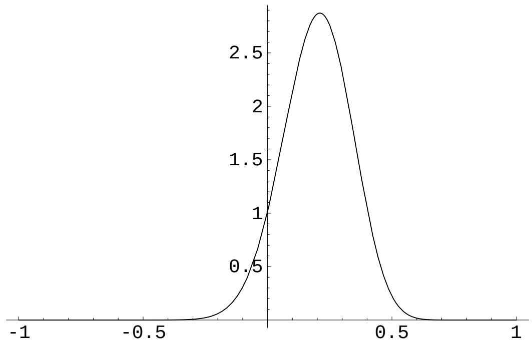

Figure B.2 Density given by Hotelling for values of $r = .40$ and df $= 48$.

The Fisher approximation. Here and in the following parts the code begins with the definition of the relationship between r and its transformation. In the present case it is simply the Fisher-z transformation

```
FisherZFromR[r_] := 1/2Log[(1 + r)/(1 - r)]
```

Next, the derivative of Z with respect to r is computed with an additional simplification of the expression for convenience. This step is presented here for completeness and will be left out for the other approaches. The step is helpful for the following change of variables.

```
FullSimplify[D[FisherZFromR[r], r]]
```

The above step results in $\frac{1}{1-r^2}$ which is inserted in the following expression

```
DerivativeOfFisherZFromR[r_] := 1/(1 - r^2)
```

Now the density of R that results from the application of the Fisher-z transformation is defined by a change of variables.

```
RDensityFisher[x_, nu_, rho_] :=
      PDF[NormalDistribution[1/2Log[(1 + rho)/(1 - rho)],
          1/(Sqrt[nu - 1])],FisherZFromR[x]]
          DerivativeOfFisherZFromR[x]
```

Note, that the parameters of the normal distribution in MATHEMATICA are the expected value and the standard deviation. To compute the expected value and variance of this distribution, respectively, the following expressions are used to integrate over the interval [-0.99999, 0.99999] using the density of R as given above.

```
ExpectationOfFishersR[nu_, rho_] :=
      NIntegrate[x RDensityFisher[x, nu, rho], {x, -0.99999, 0.99999}]
```

With the following function call the expected value is computed for the values defined in the general part.

```
ExpectationOfFishersR[df, ActualRho]
```

For the given example a value of 0.19607 will be returned. It is now interesting to compare this value with the one resulting from using Hotelling's density for computation. The value for the latter distribution was 0.198047 so that a simulation procedure employing the Fisher approximation will generate r values that are too small in expected value!

Accordingly, the following two expressions can be used to compute the variance of the distribution.

```
SecondMomentOfFishersR[nu_, rho_] :=
    NIntegrate[x^2 RDensityFisher[x, nu, rho], {x, -0.99999, 0.99999}]

VarOfFishersR[nu_, rho_] :=
  SecondMomentOfFishersR[nu, rho] - (ExpectationOfFishersR[nu, rho])^2
```

The following function call returns a value of 0.0189376 for the variance of the distribution which is larger than the value for Hotelling's density which was 0.0188894.

```
VarOfFishersR[df, ActualRho]
```

The Harley approximation. The code for this and the following approximation is structurally identical to the Fisher approximation, so it will not be annotated.

```
HarleysTFromR[r_, nu_, rho_] :=
    (r Sqrt[-nu (-2 + rho^2)])/(Sqrt[2 - 2 r^2])

DerivativeOfHarleysTFromR[r_, nu_, rho_] :=
    (Sqrt[nu - (nu rho^2)/ 2])/( (1 - r^2)^(3/2))

HarleyDelta[nu_, rho_] := Sqrt[(1 + 2 nu) rho^2/(2 - rho^2)]

RDensityHarley[x_, nu_, rho_] :=
     DensityNoncentralT[HarleysTFromR[x, nu, rho], nu,
     HarleyDelta[nu, rho]]DerivativeOfHarleysTFromR[x, nu, rho]

ExpectationOfHarleysR[nu_, rho_] :=
     NIntegrate[x RDensityHarley[x, nu, rho], {x, -0.99999, 0.99999}]

ExpectationOfHarleysR[df, ActualRho]

SecondMomentOfHarleysR[nu_, rho_] :=
     NIntegrate[x^2 RDensityHarley[x, nu, rho],
```

```
     {x, -0.99999, 0.99999}]

VarOfHarleysR[nu_, rho_] :=
     SecondMomentOfHarleysR[nu, rho] -
     (ExpectationOfHarleysR[nu, rho])^2

VarOfHarleysR[df, ActualRho]
```

The Samiuddin-Kraemer approximation.

```
KraemersTFromR[r_, nu_, rho_] :=
     Sqrt[nu] (r - rho)/Sqrt[(1 - r^2) (1 - rho^2)]

DerivativeOfKraemersTFromR[r_, nu_, rho_] :=
    (Sqrt[nu] (-1 + r rho) (-1 + rho^2))/
    (((-1 + r^2) (-1 + rho^2))^(3/2))

RDensityKraemer[x_, nu_, rho_] :=
    DensityStudentT[KraemersTFromR[x, nu, rho], nu]
    DerivativeOfKraemersTFromR[x, nu, rho]

ExpectationOfKraemersR[nu_, rho_] :=
    NIntegrate[x RDensityKraemer[x, nu, rho], {x, -0.99999, 0.99999}]

ExpectationOfKraemersR[df, ActualRho]

SecondMomentOfKraemersR[nu_, rho_] :=
    NIntegrate[x^2 RDensityKraemer[x, nu, rho],
    {x, -0.99999, 0.99999}]

VarOfKraemersR[nu_, rho_] :=
    SecondMomentOfKraemersR[nu, rho] -
    (ExpectationOfKraemersR[nu, rho])^2

VarOfKraemersR[df, ActualRho]
```