Further details

Pieter Vermeesch

02/03/2021

This document compiles the answers to some student questions that are best addressed by numerical simulation.

1 Degrees of freedom

The degrees of freedom of a problem represents the number of independent ways in which a system can vary. In most cases, the degrees of freedom represents the number of measurements, adjusted for the number of parameters. Perhaps the clearest illustration of this concept is given in Section 9.2 of the notes.

In equation 9.2, we use n measurements to estimate the one-sample t-statistic t. But to do so we also have to calculate $\bar{x}$. So there is some redundancy in the system, which is reducing the apparent dispersion of the t-statistic. To account for this, we subtract one degree of freedom, in exactly the same way as the Bessel correction, which is briefly discussed at the end of Chapter 7. See Wikipedia for a derivation of the Bessel correction.

In the two-sample t-test (Equation 9.4), we have $n_1 + n_2$ measurements, and have to estimate two parameters, $\bar{x}_1$ and $\bar{x}_2$. Hence the number of degrees of freedom is $n_1 + n_2 - 2$.

It is a useful exercise to simulate the t-distribution on your computer. For example, for the one-sided t-test (Equation 9.2):

```
ns <- 1000      # number of samples
nv <- 2         # number of values per sample
# 1000 samples of 2 values drawn from a standard normal distribution:
obs <- matrix(rnorm(nv*ns),nrow=ns,ncol=nv)
tstat <- rep(NA,ns)  # initialise the t-statistic
for (i in 1:ns){     # loop through the samples
    tstat[i] <- sqrt(nv)*mean(obs[i,])/sd(obs[i,])  # equation 9.2 of the notes
}
# predicted quantiles of the t-distribution with nv-1 degrees of freedom:
pred1 <- qt(seq(from=0,to=1,length.out=ns),df=nv-1)
# predicted quantiles of the t-distribution with nv degrees of freedom:
pred2 <- qt(seq(from=0,to=1,length.out=ns),df=nv)
# plot the empirical cumulative distribution function of the 1000 t-statistics:
plot(ecdf(tstat),verticals=TRUE,pch=NA,col='blue',xlim=c(-5,5),xlab='t',main='')
# add the predicted distribution:
lines(ecdf(pred1),verticals=TRUE,pch=NA,col='black')
# add the second predicted distribution:
lines(ecdf(pred2),verticals=TRUE,pch=NA,col='red')
legend('topleft',legend=c('measured','n-1 d.o.f','n d.o.f'),
       lty=1,col=c('blue','black','red'))
```

which produces Figure 1. If you want you can modify this code for the two-sample case to verify that this requires $n_1 + n_2 - 2$ degrees of freedom.

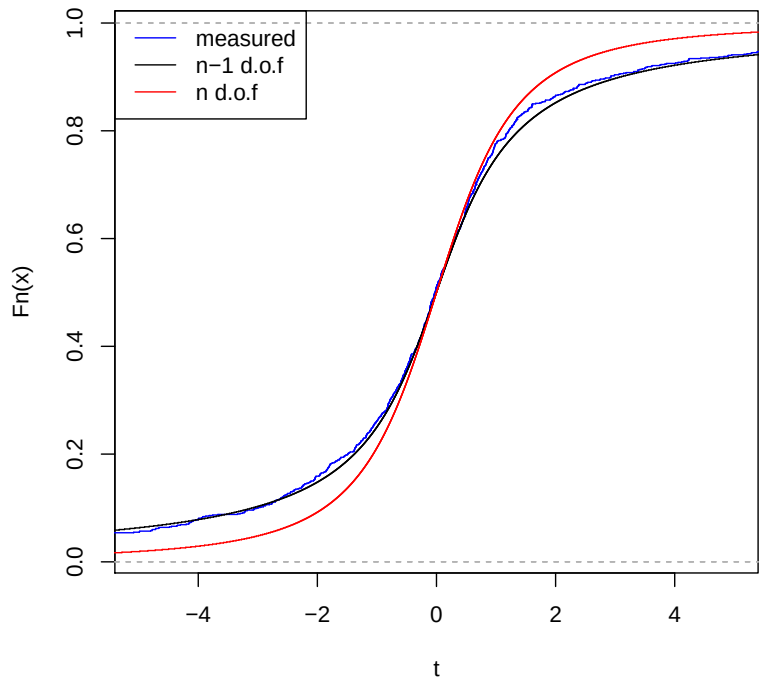

Figure 1: blue: ECDF of 1000 simulated t-statistics calculated by averaging $n = 2$ values; black: theoretical CDF of a t-distribution with 1 degree of freedom; red: theoretical CDF of a t-distribution with 2 degrees of freedom. $n - 1 = 1$ d.o.f clearly works better.

2 Kolmogorov-Smirnov

The `ks.test` function has an optional argument called `alternative`, which can take on three values:

1. `alternative="two.sided"` implements Equation 9.11 of the notes, in which the Kolmogorov-Smirnov test statistic D is defined as the absolute value of the greatest vertical distance between two empirical cumulative distribution functions:
$$D = \max_z |F_x(z) - F_y(z)|$$
2. `alternative="greater"` only looks at the maximum positive difference:
$$D^+ = \max_z (F_x(z) - F_y(z))$$
3. `alternative="less"` only looks at the maximum negative difference:
$$D^- = \max_z (F_y(z) - F_x(z))$$

The two-sided case is by far the most commonly used type of Kolmogorov-Smirnov test, and is the only one that you are expected to use in GEOL0061.

However, for the sake of completeness, let us illustrate the difference between the three different options by simulating 1000 sets of samples from two normally distributed variables (A and B), where both A and B were drawn from a normal distribution with:

1. $\mu_A = \mu_B = 10$ and $\sigma_A = \sigma_B = 3$
2. $\mu_A = 10$ and $\mu_B = 9$ and $\sigma_A = \sigma_B = 3$
3. $\mu_A = 10$ and $\mu_B = 11$ and $\sigma_A = \sigma_B = 3$

```
par(mfrow=c(3,4),mar=c(3,3,1,0),mgp=c(1.5,0.5,0))
ns <- 500  # number of samples
```

```r
ni <- 1000 # number of iterations
mA <- rep(10,3)  # mean of distribution A
mB <- c(10,9,11) # mean of distribution B
for (k in 1:3){
    Dm <- rep(Inf,ni)  # initialise vector of D- values
    DM <- rep(-Inf,ni) # initialise vector of D+ values
    D <- rep(Inf,ni)   # initialise vector of D values
    for (j in 1:ni){
        A <- rnorm(ns,mean=mA[k],sd=3) # draw a sample from distribution A
        B <- rnorm(ns,mean=mB[k],sd=3) # draw a sample from distribution A
        D[j] <- ks.test(A,B,alternative='two.sided')$statistic
        Dm[j] <- ks.test(A,B,alternative='less')$statistic
        DM[j] <- ks.test(A,B,alternative='greater')$statistic
    }
    plot(ecdf(A),verticals=TRUE,pch=NA,col='blue',xlim=range(c(A,B)),main='')
    legend('topleft',legend=c('A','B'),lty=rep(1,2),col=c('red','blue'),bty='n')
    lines(ecdf(B),verticals=TRUE,pch=NA,col='red')
    hist(Dm,xlab=expression('D'^'-'),main=''); rug(Dm)
    hist(DM,xlab=expression('D'^'+'),main=''); rug(DM)
    hist(D,xlab='D',main=''); rug(D)
}
```

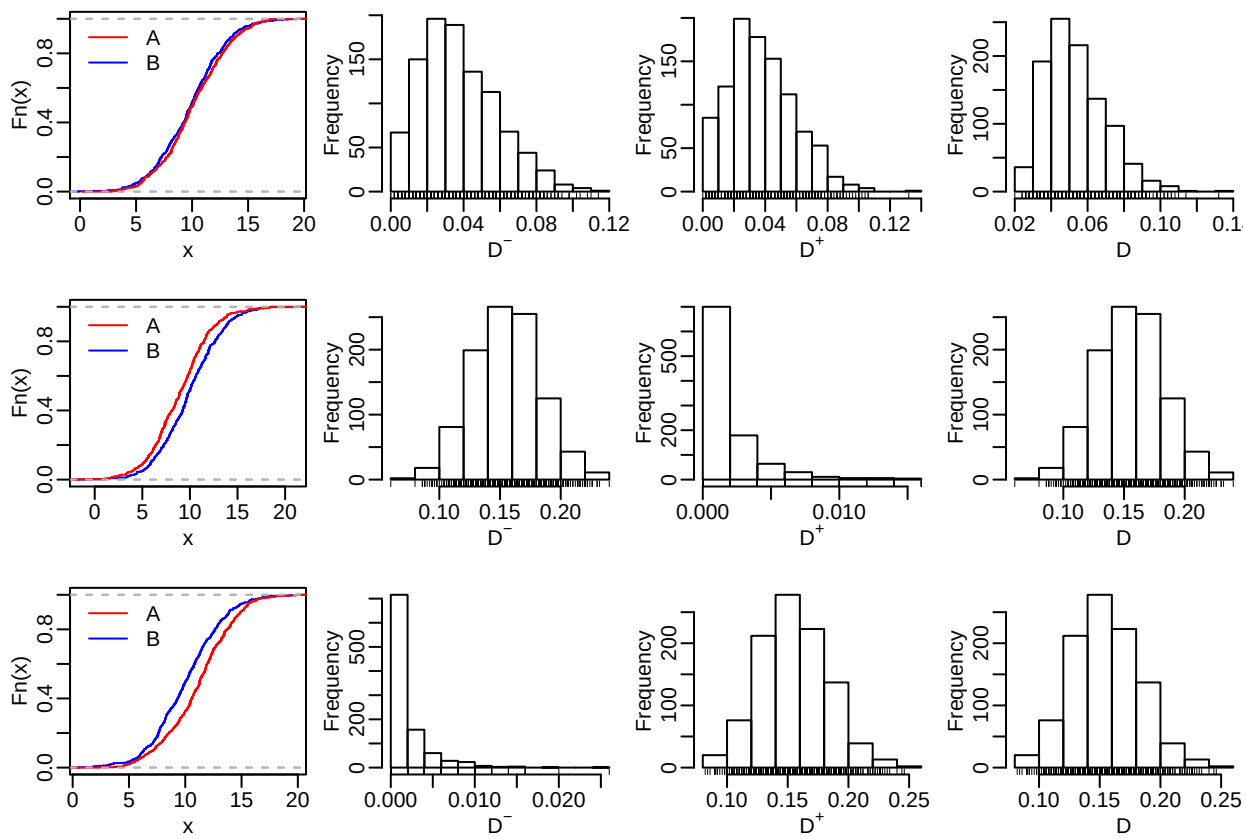

Figure 2: From left to right: the ECDFs of representative samples of distribution A (red) and B (blue); and the sampling distribution of 1000 simulated D^+, D^- and D-statistics. From top top bottom: the outcomes for $\mu_A = \mu_B$, $\mu_A < \mu_B$ and $\mu_A > \mu_B$.

Figure 2 shows that, of the two samples were not drawn from the same population, there is a distinct

asymmetry between the expected D^+ and D^- distributions.