

```
####
#### Statistical hypothesis tests in R
####
#### Author: Stefan Evert
#### Modified: Mon Apr 18 18:24:17 2005 (evert)
####

##
## Introduction to hypothesis tests: binomial test
##

# fair die (p=1/6) vs. tampered die (p=1/2): compare number of sixes among 12 throws
# -> combined plot of binomial distributions with n=12 and p=1/6 vs. p=1/2
k <- 0:12
p.fair <- dbinom(k, size=12, prob=1/6)
p.tampered <- dbinom(k, size=12, prob=1/2)
table <- rbind(p.fair, p.tampered)
colnames(table) <- 0:12
print(round(table, digits=4))

barplot(table, beside=T, legend.text=c("p=1/2", "p=1/6"), xlab="number of sixes",
ylab="probability", main="Fair die vs. tampered die")
# -> assume tampered die for k >= 4 sixes

# probability of type I error (the line below computes P(X > 3) which is equivalent to P
(X >= 4))
pbinom(3, size=12, prob=1/6, lower.tail=FALSE)
# probability of type II error
pbinom(3, size=12, prob=1/2)

# taking prior probabilities into account (assuming that we pick a die at random
# from a "population" containing a proportion of 10% of tampered dice)
table[1,] <- table[1,] * 0.9
table[2,] <- table[2,] * 0.1
barplot(table, beside=T, legend.text=c("p=1/2", "p=1/6"), xlab="number of sixes",
ylab="total probability", main="Fair die vs. tampered die")
# -> decision rule is now: assume tampered die for k >= 6 sixes

# total probabilities of type I and type II errors
0.9 * pbinom(5, size=12, prob=1/6, lower.tail=FALSE)
0.1 * pbinom(5, size=12, prob=1/2)

# how binomial probabilities P(X=k) depend on sample size n:
# plot binomial distributions for p=1/7 and n=20 vs. n=100
k <- 0:30
p.20 <- dbinom(k, size=20, prob=1/7) # k > 20 will return probability = 0
p.100 <- dbinom(k, size=100, prob=1/7)
table <- rbind(p.20, p.100)
colnames(table) <- 0:30
barplot(table, beside=T, legend.text=c("n=20", "n=100"), xlab="number of successes",
ylab="probability", main="Binomial distributions for p=1/7")

# two-sided binomial test for null hypothesis p=1/7; with sample size n=50
k <- 0:24
p <- dbinom(k, size=50, prob=1/7)
names(p) <- 0:24
mp <- barplot(p, ylim=c(0,0.2), xlab="number of successes", ylab="probability",
main="Binomial distribution for p=1/7 and n=50")
# show expectation value as red line; since barplot computes its own x-coordinates (which
we can't adjust
# with the xlim parameter), we have to find the x-value corresponding to the expected
```

```

value 50/7;
# the barplot() function returns the x-midpoints of the bars drawn as a vector (assigned
to mp);
# because the bars are labelled 0,1,2,... in our case, the x-coordinate mp[1] corresponds
to k=0,
# mp[2] corresponds to k=1, etc.; therefore, in order to rescale the expectation  $k = 50/7$ 
= 7 1/7,
# we have to compute a weighted mean of mp[8] (k=7) and mp[9] (k=8)
x.expected <- (1 - 1/7) * mp[8] + 1/7 * mp[9]
abline(v=x.expected, lwd=2, col="black")
text(x.expected + 0.2, 0.17, labels="expected value", pos=4, srt=90)

# significance value of k=3 successes (two-sided binomial test)
sum(dbinom(0:3, size=50, prob=1/7)) + sum(dbinom(12:50, size=50, prob=1/7))
# pbinom() might produce more accurate results than large sums
pbinom(3, size=50, prob=1/7) + pbinom(11, size=50, prob=1/7, lower.tail=FALSE)

# R has a built-in function for the binomial test
binom.test(3, n=50, p=1/7, alternative="two.sided")
# now we just have to compare the computed p-value to our pre-defined significance level

# one-sided test gives lower p-values and may reject  $H_0$  when two-sided test doesn't
binom.test(3, n=50, p=1/7, alternative="less")

##
## Power function of the binomial test
##

# draw power function of one-sided binomial test for  $H_0: p \leq 1/7$  against  $H_1: p > 1/7$ 
# comparing its power for sample sizes  $n=20$  and  $n=100$  at significance level  $\alpha=0.05$ 

# first, compute rejection criterion  $k \geq L$  for both sample sizes
p <- pbinom(2:9, size=20, prob=1/7, lower.tail=FALSE)
names(p) <- 3:10
print(round(p, digits=4))
#  $P_0(X \geq k)$  is below significance level ( $\alpha=0.05$ ) for  $k \geq 7 \rightarrow L = 7$ 
p <- pbinom(14:29, size=100, prob=1/7, lower.tail=FALSE)
names(p) <- 15:30
print(round(p, digits=4))
#  $P_0(X \geq k)$  is below significance level ( $\alpha=0.05$ ) for  $k \geq 21 \rightarrow L = 21$ 

# now compute probability of rejection  $P(X \geq L)$  for different values of the
# true success probability p
p <- seq(from=0, to=1, by=0.01)
# recent versions of R also accept a vector argument for the prob parameter
# (but the first parameter k should then be a simple scalar!)
f.20 <- pbinom(7-1, size=20, prob=p, lower.tail=FALSE)
f.100 <- pbinom(21-1, size=100, prob=p, lower.tail=FALSE)

# plot the graphs of the first power function
plot(p, f.20, xlim=c(0,1), ylim=c(0,1), type="l", lwd=2, col="blue", xlab="true value of
parameter p", ylab="probability of rejection", main="Power function of binomial test for
 $p \leq 1/7$  against  $p > 1/7$ ")
# show limits for power function on x-axis and y-axis
abline(h=0, lwd=2, col="black")
abline(h=1, lwd=2, col="black")
abline(v=0, lwd=2, col="black")
abline(v=1, lwd=2, col="black")
# border between  $H_0$  and  $H_1$ 
lines(c(1/7, 1/7), c(0,1), col="black")
text(1/7, 0.95, labels=expression(H[1]), pos=4)

```

```

text(1/7, 0.95, labels=expression(H[0]), pos=2)
# type I error
lines(c(p[14], p[14]), c(0.0, f.20[14]), lwd=2, col="red")
text(p[15], f.20[14], pos=4, labels="<- type I error")
# significance level alpha=0.05
lines(c(0,1/7), c(0.05,0.05), lwd=2, col="black")
text(1/14, 0.05, pos=3, labels="significance\nlevel")
# type II error
lines(c(p[41], p[41]), c(f.20[41], 1.0), lwd=2, col="red")
text(p[41], 0.9, pos=2, labels="type II error ->")
# add graph for second power function and legend
lines(p, f.100, type="l", lwd=2, col="darkgreen")
legend(0.95, 0.05, xjust=1, yjust=0, lwd=2, col=c("blue", "darkgreen"), legend=c("n =
20", "n = 100"))

##
## Hypothesis tests for normal distributions
##

# z-scores for IQ values, assuming a normal distribution; H_0: mu=100 and sigma=15
# random sample of n students (from different population, so H_0 should ideally be
rejected)
n <- 10
x <- rnorm(n, mean=110, sd=15)
print(round(sort(x), digits=2))
x.bar <- mean(x)
print(x.bar)

# distribution of random variable X.bar under H_0 ("sampling distribution")
mu.bar <- 100
sigma.bar <- 15 / sqrt(n)

# significance of observed value X.bar is the two-sided probability of the same
# or a greater deviation from the expected value mu=100;
# because the normal distribution is symmetric, we can just multiply the one-sided
# probability P(X.bar >= x.bar) by two;
# note that pnorm(x, mean=..., sd=..., lower.tail=FALSE) == 1 - pnorm(X, mean=...,
sd=...)
# because the normal distribution is a continuous distribution with density function
2 * pnorm(x.bar, mean=mu.bar, sd=sigma.bar, lower.tail=FALSE)
# note that this may or may not reject the null hypothesis (depending on random numbers)

# for the one-sided test (H_1: mu > 100), we obtain a smaller significance value
pnorm(x.bar, mean=mu.bar, sd=sigma.bar, lower.tail=FALSE)

# traditionally, the mean x.bar is standardised (under H_0) and is reported as z-score
z.score <- (x.bar - mu.bar) / sigma.bar
print(z.score)

# under H_0, the z-score follows a standard normal distribution (default for pnorm()
function)
2 * pnorm(z.score, lower.tail=FALSE)

# a naive application of the chi^2 distribution: standardised squared error
x.std <- (x - 100) / 15 # gives standard normal distributions under H_0
chi.sq <- sum(x.std ^ 2) # chi-squared distribution with df=n under H_0
print(chi.sq)

# convert chi.sq value to significance (p-value); corresponds to two-sided test
pchisq(chi.sq, df=n, lower.tail=FALSE)

```

```
# Student's t test: t statistic is computed from mean and sample variance
s2 <- var(x)
t <- (x.bar - 100) / sqrt(s2 / n)
print(t)

# convert t value to significance; corresponds to one-sided test (H_1: mu > 100)
pt(t, df=n-1, lower.tail=FALSE)

# compare sample standard deviation (or variance) to assumed s.d.=15
sqrt(s2)

# R has a built-in function for the t test (also for the two-sided test)
t.test(x, mu=100, alternative="greater")
t.test(x, mu=100, alternative="two.sided")

# another example: body heights of American women aged 30-39
data(women)
# convert heights given in inches to centimeters
x <- women$height * 2.54
print(x)

# data from the IMS suggests that the average modern young girl is about 170 cm tall;
# in order to test this claim for American women, we assume that heights are normally
# distributed (because of the central limit theorem) with unknown standard deviation
# and perform a t-test for the null hypothesis mu=170
t.test(x, mu=170, alternative="two.sided")
```