Stat 311: HW 10, not due, solutions to be posted before final

Fritz Scholz

1. Section 13.4, problem 1. Basically you test the hypothesis $p_1 = \ldots = p_8 = 1/8$, where p_i denotes the probability that the winner comes from starting position i. This is based on assigning the horses at random to the starting positions and the hypothesis assumption that starting position has no influence on who the winner is. Thus the winner has equal chance of coming from any of the 8 starting positions. Since no significance level α is specified, just compute the p-value and comment on the strength of the evidence against the hypothesis.

2. Section 13.4, problem 3. Again, no α is specified. Thus compute the p-value and comment on the strength of the evidence against the hypothesis that the cell probabilities in (a) are correct.

3. Section 13.4, problem 5. Note that $\log_{10}(x)$ gives you that number y such that $10^y = x$, e.g., $\log_{10}(1000) = 3$ since $10^3 = 1000$. Furthermore, we have

$$\log_{10}(x) \geq 0 \qquad \text{for } x \geq 1 \tag{1}$$

and

$$y = \log_{10}(x_1 \cdot x_2 \cdot \ldots \cdot x_n) = \log_{10}(x_1) + \log_{10}(x_2) + \ldots + \log_{10}(x_n) = y_1 + y_2 + \ldots + y_n \tag{2}$$

since

$$10^{y_1+y_2+\ldots+y_n} = 10^{y_1} \cdot 10^{y_2} \cdot \ldots 10^{y_n} = x_1 \cdot x_2 \cdot \ldots \cdot x_n = 10^y$$

Use the properties (1) and (2) to do part (a). In R the command `p <- log10(1+1/(1:9))` would give you the vector of required cell probabilities. You can also check `sum(p)`.

Benford's law (see `http://en.wikipedia.org/wiki/Benford%27s_law`) is quite useful in detecting fraudulent activities when numbers are just made up, such as in accounting when cooking the books, or in faking election results. The latter issue was most recently examined w.r.t. the election results in Iran.
`http://blog.jgc.org/2009/06/benfords-law-and-iranian-election.html`

Since no α is specified, work with the p-value to assess the evidence.

4. Section 14.6, problem 7. Rather than just calling `cor.test(sister,brother,conf.level=0.9)` for appropriate data vectors `sister` and `brother`, write yourself a function (using the steps on slides 27-28 in Ch. 14)

```
Conf.Int <- function(x,y,conf.level){
....
}
```

that computes $\hat{\rho}$, then computes $\hat{\zeta}$, then computes the interval end points `zeta.L` and `zeta.U` for ζ, and from that computes the interval endpoints `rho.L` and `rho.U` for ρ and returns as output `c(rho.L,rho.U)`. Compare the results from using

$$\texttt{Conf.Int(sister,brother,conf.level} = 0.9)$$

with that when using

$$\texttt{rho.test(sister,brother,conf.level} = 0.9)$$

You may use the fact that `cor(x,y)` returns the sample correlation coefficient for the data vector `x` and `y`.