Stat 430 Homework 1

due: October 1, 2009

This (and future) homeworks will require you have access to `R`. You can install `R` on your own computer, but `R` is also availabe on the thin clients in the various computer labs of Snedecor hall. As a student you can also connect to the Statistics Department's terminal server. I will indicate how to do this in Windows. Open Remote Desktop Connection (Start → Program Files → Accessories) and open a connection to `ts.stat.iastate.edu` with your ISU username/password. This is the same server the thin clients use.

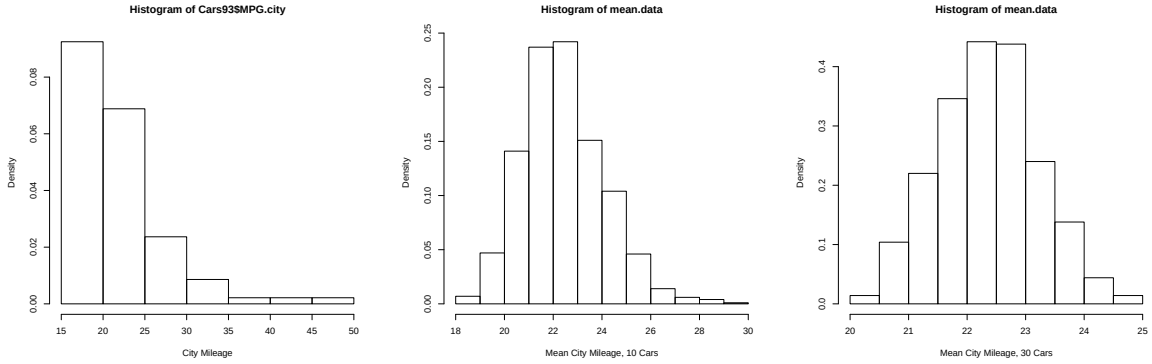
When turning in the homework, please email code that *runs*. The answers should either be printed out by the code (see function `pdf()` for writing pdf graphics files) or you must also submit another *single* document (paper copy or forwarded in email) that neatly displays *all* the answers obtained from the code. In all, you should send or deliver to me *at most two files* for this homework. Please comment your code and use `print()` statements to explain any output your code produces.

1. The MASS package contains a dataset `Cars93`, which you can load with the command `data(Cars93)`. This dataset lists the types of cars on sale in the USA in 1993.
   (a) There are several types or categories of cars, listed under the `Type` column. Find the cheapest car in each category.
   (b) Compute the mean horsepower and mean city mileage (MPG) for each type.
   (c) What other quantitative variable in this dataset, other than highway mileage, shows the highest absolute correlation with city mileage? Report the variable and the maximum correlation.
   (d) Use `write.table()` to save the manufacturer, model, type, price and mileage data for cars of US origin to a file. Load the data from the file and report the output of the `head()` command applied to the loaded dataset.
   (e) Plot a histogram of city mileage. Does it look normally distributed?
   (f) Use the command `sample()` to sample $n = 10$ cars and compute the mean city mileage. Now, repeat $B = 999$ times, collecting all $1,000$ means, and plot a histogram of sample means. Does the result normal? If not, does it look normal if $n = 30$ and $B = 1000$?

Solution: There are many ways to write the `R` code. I have made some attempt to make the following efficient. If you are interested in learning `R`, you can learn some tricks from studying the code.

`R` Code:

```
library(MASS)         # load library
data(Cars93)          # load data, so accessible as Cars93 now
cat("1(a)\n")
```



(a) Raw data (b) $n = 10$ (c) $n = 30$

Figure 1: Histogram of (a) city mileage or mean city mileage for 1000 random samples of size (b) $n = 10$ or (c) $n = 30$.

```
# tapply applies function (min) to selected elements of Min.Price
# selection is recursively on the levels of Type
print(tapply(Cars93$Min.Price, Cars93$Type, min))
cat("1(b)\n")
cat("Mean Horsepower\n==============\n")
print(tapply(Cars93$Horsepower, Cars93$Type, mean))
cat("Mean City Mileage\n================\n")
print(tapply(Cars93$MPG.city, Cars93$Type, mean))
cat("1(c)\n")
# select numeric columns of Cars93 only
Cars93.num <- Cars93[,as.vector(lapply(as.list(Cars93), is.numeric), mode="logical")]
# deselect the MPG columns, since we're looking for OTHER high correlations
Cars93.num <- Cars93.num[,names(Cars93.num) != "MPG.city" & names(Cars93.num) != "MPG.highway"]
# compute correlation for each column in Cars93.num
Cars93.cor <- cor(Cars93$MPG.city, Cars93.num, use="pairwise.complete.obs")
i <- which.max(abs(Cars93.cor))   # return location of maximum absolute correlation
print(Cars93.cor[1,i])
cat("Note: reported maximum absolute correlation\n")
cat("1(d)\n")
write.table(x=Cars93[Cars93$Origin=="USA",
   c("Manufacturer","Model","Type","Price","MPG.city","MPG.highway")],
   file="USCars.Rtxt")
US.cars <- read.table("USCars.Rtxt")
print(head(US.cars))
cat("1(e)\n")
pdf("CityMileage.pdf")
hist(Cars93$MPG.city, xlab="City Mileage", freq=F)
dev.off()
cat("See file CityMileage.pdf\n")
cat("The sample density looks far from normal, with a long right tail and a truncated left tail\n")
cat("1(f)\n")
# replicate repeats, often something random, n times
mean.data <- replicate(n=1000, mean(sample(Cars93$MPG.city, size=10, replace=F)))
pdf("MeanCityMileage10.pdf")
hist(mean.data, xlab="Mean City Mileage, 10 Cars", freq=F)
dev.off()
cat("See file MeanCityMileage10.pdf\n")
cat("The sample density does not look completely normal because there is still a longer right tail.\n")
mean.data <- replicate(n=1000, mean(sample(Cars93$MPG.city, size=30, replace=F)))
pdf("MeanCityMileage30.pdf")
hist(mean.data, xlab="Mean City Mileage, 30 Cars", freq=F)
dev.off()
cat("See file MeanCityMileage30.pdf\n")
cat("It looks a little more normal, because the skew seems to have disappeared.\n
However, the tails may be truncated, because the sample is large, nearly 1/3, of the population.\n
Thus, the means are more similar, less variable than expected.\n")
```

Output.

```
1(a)
Compact   Large Midsize   Small  Sporty     Van
    8.5    17.5    12.4     6.7     9.1    13.6
1(b)
Mean Horsepower
===============
 Compact    Large  Midsize    Small   Sporty      Van
131.0000 179.4545 173.0909  91.0000 160.1429 149.4444
Mean City Mileage
=================
 Compact    Large  Midsize    Small   Sporty      Van
22.68750 18.36364 19.54545 29.85714 21.78571 17.00000
1(c)
    Weight
-0.8431385
Note: reported maximum absolute correlation
1(d)
   Manufacturer      Model    Type Price MPG.city MPG.highway
6         Buick    Century Midsize  15.7       22          31
7         Buick    LeSabre   Large  20.8       19          28
8         Buick Roadmaster   Large  23.7       16          25
9         Buick    Riviera Midsize  26.3       19          27
10     Cadillac    DeVille   Large  34.7       16          25
11     Cadillac    Seville Midsize  40.1       16          25
1(e)
See file MeanCityMileage10.pdf
The sample density does not look completely normal because there is still a longer right tail.
See file MeanCityMileage30.pdf
It looks a little more normal, because the skew seems to have disappeared.
However, the tails may be truncated, because the sample is large, nearly 1/3, of the population.
Thus, the means are more similar, less variable than expected.
```

2. Design and implement a simulation study to confirm the claim that if random variable $X \sim$Poisson(λ) represents the number of events, and each event is independently marked with probability p, then Y, the number of marked events follows a Poisson(λp) distribution.

   (We have now learned how to formally test whether the simulation data agrees with a Poisson(λp) distribution. I am not asking for a formal goodness-of-fit. Visual confirmation of a match is sufficient.)

Solution:

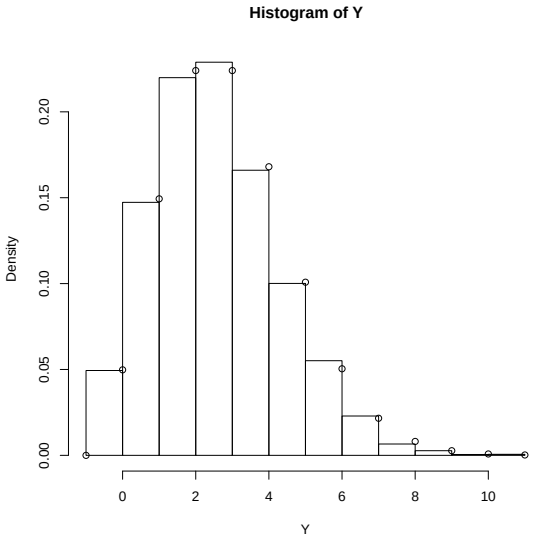


Figure 2: Similarity between random variables generated from Poisson(λ) and marked independently with probability p (histogram) and the theoretical Poisson(λp) distribution (circles).

R Code.

```
lambda <- 10
p <- 0.3
N <- 10000
cat("I have picked parameters lambda = ", lambda, "and p =", p,
   ", and I will sample", N, "times.\n")
X <- rpois(n=N, lambda=lambda)
Y <- mapply(rbinom, n=rep(1, N), size=X, prob=p)
pdf("ApproximatePoisson.pdf")
x <- (-1):max(Y)
hist(Y, xlab="Y", freq=F, breaks=x)
points(x=x, y=dpois(x, lambda=p*lambda))
dev.off()
cat("Please see figure ApproximatePoisson.pdf,
   which shows a close match between theory and simulation.\n")
```

with output

```
I have picked parameters lambda =  10 and p = 0.3 , and I will sample 10000 times.
Please see figure ApproximatePoisson.pdf, which shows a close match between theory and simulation.
```

3. The activity of web servers can be measured by counting incoming http requests. For a certain server, the counts in consecutive five-minute intervals may be regarded (approximately) as repeated independent observations from a normal distribution. Suppose the mean five-minute count for this server is $1,200$ and the standard deviation is 35. For that server, let Y represent a five-minute count and let $\bar{Y}$ represent the mean of six five-minute counts. Find $P(1175 \leq Y \leq 1225)$ and $P(1175 \leq \bar{Y} \leq 1225)$ and compare the two. Does the comparison indicate that counting for thirty minutes and dividing by six would tend to give a more precise result than merely counting for a single five-minute interval? How?

Solution:

The relevant distributions are

$$Y \sim N(1200, 35^2) \qquad\qquad \bar{Y} \sim N(1200, 35^2/6)$$

R code:

```
cat("Probability for Y:", pnorm(1225, mean=1200, sd=35) - pnorm(1175, mean=1200, sd=35), "\n")
cat("Probability for Y.bar:", pnorm(1225, mean=1200, sd=35/sqrt(6))
   - pnorm(1175, mean=1200, sd=35/sqrt(6)), "\n")
```

with output

```
Probability for Y: 0.5249495
Probability for Y.bar: 0.9198188
```

Clearly, $\bar{Y}$ provides a more precise estimate (smaller confidence intervals) because its sampling variance is smaller than the sampling variance of Y from a single interval.

4. It is well known that one in ten users of a certain type of software call technical support for assistance. If a company sells 15 copies of the software and receives no calls, what would be their estimate of the proportion of *their clients* who call technical support? Construct a 95% confidence interval for this estimate and justify your method of construction. What is wrong with the substitution method for this application?

Solution: The maximum likelihood estimate is $\hat{p} = 0$. The confidence interval is

$$0 \pm 1.96\sqrt{\frac{p(1-p)}{15}}$$

but since p is unknown for the population of clients for this company, we'll need to substitute in some value. We could use the conservative intervals, setting $p = 1/2$:

$$\pm 0.2530349$$

We could use the current best estimate of the population proportion $p = 0.1$:

$$\pm 0.1518209$$

We cannot use the substitution method, because $\hat{p} = 0$ would produce zero-length intervals. We prefer the population proportion approach unless we believe the clients of this company are very different from the clients across the industry. Also, since $p \geq 0$, the lower limit should be truncated at 0.

5. Derive the maximum likelihood estimates of λ and p if you observe n independent pairs (X_i, Y_i) (note X_i is not independent of Y_i), where $X_i \sim$Poisson(λ) and Y_i is the number of marked (independently with probability p) events in X_i. (See problem 2.)

Solution: Both are discrete random variables, and for any pair (X, Y), the joint probability is

$$P(X, Y) = P(Y|X)P(X)$$

where $P(Y|X)$ is Binomial(X, p) and $P(X)$ is Poisson(λ). Since (X_i, Y_i) are independent of other pairs, the joint probability of all data is

$$P\left[(X_1, Y_1), \ldots, (X_n, Y_n)\right] = \prod_{i=1}^{n} P(X_i, Y_i) = \prod_{i=1}^{n} P(Y_i|X_i) \prod_{i=1}^{n} P(X_i)$$

which yields the likelihood when viewed as a function of the parameters λ and p. Because the likelihood factors into two separate (not independent) parts where one involves a product of Binomial probabilities and the other a product of Poisson probabilities, the maximum likelihood calculations for λ and p effectively separately, and the calculations will look familiar. The log likelihood is

$$\begin{aligned}
l(\lambda, p) &= \sum_{i=1}^{n} \left[\ln \binom{x_i}{y_i} + y_i \ln p + (x_i - y_i) \ln(1-p) - \lambda + x_i \ln \lambda - \ln x_i! \right] \\
\frac{dl}{d\lambda} &= -n + \frac{\sum_{i=1}^{n} x_i}{\lambda} \\
\hat{\lambda} &= \frac{1}{n} \sum_{i=1}^{n} x_i \\
\frac{dl}{dp} &= \frac{\sum_{i=1}^{n} y_i}{p} - \frac{\sum_{i=1}^{n} (x_i - y_i)}{1-p} \\
\hat{p} &= \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i}
\end{aligned}$$

6. [added] The ABO blood locus is a gene on chromosome 9 where one of three gene variants (called alleles) are possible: A, B, or O. Humans have two copies of every gene, so they have two ABO alleles. Individuals who have two A's, denoted by genotype AA, are blood type A, but so are individuals with genotype AO. The correspondence between blood types and genotypes at the ABO locus are shown in the table.

Blood Type	Type A	Type B	Type O	Type AB
Genotypes	AA or AO	BB or BO	OO	AB
Probability	$p_A^2 + 2p_Ap_O$	$p_B^2 + 2p_Bp_O$	p_O^2	$2p_Ap_B$
Armenia	0.31	0.50	0.13	0.06
Data	52	37	11	0

The third row of the table proposes a model for the probability of each category based on the population probability of each allele: p_A, p_B, and p_O. The fourth row presents the known genotype proportions in the Armenian population. The last row presents some data on the number of individuals in each category from a sample of size $n = 100$

(a) Use the data to compute the p-value for testing the hypothesis $H_O : P_A = 0.31$, where P_A is the probability of type A individuals.

(b) Perform a goodness-of-fit test to determine if the observed data are consistent with the Armenian population proportions. Compare the results with the above test. Which test, do you think, has more *power* to reject H_0 when it is not true. I'm not asking for a quantitative answer: guess and justify your choice with words.

(c) In the US, the allele probabilities are known to be $p_A = 0.21, p_B = 0.06$, and $p_O = 0.73$. Use the probability model to test whether the data are consistent with the US population.

(d) Use `R` function `optim` to numerically find the maximum likelihood estimates $\hat{p}_A$ $\hat{p}_B$ for this data set. Notice $\hat{p}_O = 1 - \hat{p}_A - \hat{p}_B$ need not be estimated. Also, the parameters are constrained such that $0 \leq p_A + p_B \leq 1$. It is difficult to impose this constraint directly via `optim`, so I suggest adding the following code to `optim()`'s required `fn` function that you must write:

```
# suppose p is the vector of parameters (p_A,p_B)
# the following code keeps p properly constrained
if(sum(p)>1) p <- p/(sum(p)+2e6)
p[1] <- max(p[1], 1e-6)
p[2] <- max(p[2], 1e-6)
p[1] <- min(p[1], 1-1e-6)
p[2] <- min(p[2], 1-1e-6)
```

For additional help on how to use `optim()`, please see the Examples at the bottom of the `optim()` help file (`?optim`).

Solution:

6(a) R code:

```
d <- c(52,37,11,0)    # enter data
n <- sum(d)           # sample size
p.hat <- d["A"]/n     # estimated proportion
# compute Z statistic and p-value
Z <- (p.hat - 0.31)/sqrt(0.31*(1-0.31)/n)
p.val <- 2*pnorm(-abs(Z))
cat("p-value for rejecting H0: pA = 0.31:", p.val, "\n")
```

with output

```
p-value for rejecting H0: pA = 0.31: 5.609254e-06
```

6(b) R code:

```
p <- c(0.31,0.50,0.13,0.06)      # enter model
E <- n*p                         # compute expected counts
X.2 <- sum((abs(E-d)-0.5)^2/E)   # use of Yates correction not required since I had not taught it
p.val <- 1-pchisq(X.2, df=4-1)   # compute p-value (one-sided)
cat("Statistic has value", X.2, "yielding p-value", p.val, "\n")
```

with output

```
Statistic has value 21.89620 yielding p-value 6.855696e-05
```

In general, we expect the goodness-of-fit test to be more sensitive to violation of the Armenian proportions. The test on the proportion of type A only will miss deviations in the other proportions. However, if the deviance is mostly in the proportion of type A, then the test of proportion will have more power because there are fewer degrees of freedom (the critical region is larger).

6(c) R code:

```
p.A <- 0.21                      # enter hypothesized values
p.B <- 0.06
p.O <- 0.73                      # compute category probabilities
P <- c(p.A^2 + 2*p.A*p.O, p.B^2 + 2*p.B*p.O, p.O^2, 2*p.A*p.B)
E <- n*P                         # expected values
X.2 <- sum((abs(E-d)-0.5)^2/E)   # statistic
p.val <- 1-pchisq(X.2, df=3)     # compute pvalue
cat("Statistic has value", X.2, "yielding p-value", p.val, "\n")
```

with output

```
Statistic has value 124.2883 yielding p-value 0
```

6(d) R code:

```
# function to compute negative log likelihood
nll <- function(p, d) {
   # suppose p is the vector of parameters (p_A,p_B)
   # the following code keeps p properly constrained
   if(sum(p)>1) p <- p/(sum(p)+2e6)
   p[1] <- max(p[1], 1e-6)
   p[2] <- max(p[2], 1e-6)
   p[1] <- min(p[1], 1-1e-6)
   p[2] <- min(p[2], 1-1e-6)
   print(p)
   -(d["A"]*log(p[1]^2+2*p[1]*(1-p[1]-p[2])) + d["B"]*log(p[2]^2+2*p[2]*(1-p[1]-p[2]))
      + 2*d["O"]*log((1-p[1]-p[2])) + d["AB"]*log(2*p[1]*p[2]))
}# nll

p.init <- c(0.3, 0.3)                    # choose initial values
p.est <- optim(par=p.init, fn=nll, d=d)  # optimize
cat("Parameter estimates:", c(p.est$par,1-sum(p.est$par)), "\n")
```

with output

```
[1] 0.3 0.3
[1] 0.33 0.30
[1] 0.30 0.33
[1] 0.33 0.27
...
[1] 0.3298914 0.2215625
[1] 0.3299268 0.2216130
[1] 0.3297572 0.2217587
[1] 0.3299746 0.2216327
Parameter estimates: 0.3299746 0.2216327 0.4483927
```

So, the estimates are $\hat{p}_A = 0.33, \hat{p}_B = 0.22$, and $\hat{p}_O = 0.45$.

It is not required, but you could go on to test the fit of the model using a goodness-of-fit test.

```
P <- c(p.est$par[1]^2+2*p.est$par[1]*(1-sum(p.est$par)),
   p.est$par[2]^2+2*p.est$par[2]*(1-sum(p.est$par)), (1-sum(p.est$par))^2, 2*p.est$par[1]*p.est$par[2])
E <- n*P
X.2 <- sum((abs(E-d)-0.5)^2/E)
p.val <- 1-pchisq(X.2, df=3-2) # two addl parameters estimated
cat("Statistic has value", X.2, "yielding p-value", p.val, "\n")
```

with output

```
Statistic has value 25.86112 yielding p-value 3.668846e-07
```

The above results suggest that the probability model proposed in row 3 of the table does not well-explain the data in row 5, even when optimized over parameters.