Data Manipulation Exercise

Madeline Cowen

September 15, 2016

1. Researchers will often summarize P-values in genome-wide studies by making a QQ-plot. The QQ-plot has the observed (the ones you actually computed) P-values on the y-axis vs. the expected P-values on the x-axis. For a properly calibrated test, under the null hypothesis (i.e., meaning all the SNPs are in Hardy-Weinberg equilibrium) the observed P-values will follow a uniform distribution. This means that 1% of P-values will be <0.01, 5% of P-values will be <0.05, 25% of P-values will be <0.25, etc. A QQ plot is a nice way to visualize whether the P-values indeed follow a uniform distribution.

```r
setwd("~/Documents/UCLA/Bootcamp/")
snpsDataFrame=read.table('hapmap.txt',header=TRUE)
snps=as.matrix(snpsDataFrame)
```

a. To start let's revisit our tests of Hardy-Weinberg. Go back and perform the chi-square test for Hardy-Weinberg that we did in class on all SNPs in the "hapmap_CEU_r23a_chr2_ld.txt" file. Hint: you already have the code for this… Save your P-values in a vector called "pvals".

```r
compute_chisquare=function(x){
  freq=sum(x,na.rm=TRUE)/(2.0*sum(!is.na(x)))
    cnt0=sum(x==0,na.rm=TRUE)
    cnt1=sum(x==1,na.rm=TRUE)
    cnt2=sum(x==2,na.rm=TRUE)
    obscnts=c(cnt0,cnt1,cnt2)
    #print(obscnts)
    n=sum(obscnts)
    expcnts=c((1-freq)^2,2*freq*(1-freq),freq^2)*n
    chisq=sum((obscnts-expcnts)^2/expcnts)
    return(chisq)
}

chisqs=apply(snps,1,compute_chisquare)
pvals=pchisq(chisqs,1,lower.tail=FALSE)
```

b. What proportion of P-values from the test (put the vector called "pvals") are <0.05? What proportion are <0.01? Are any <0.001?

```r
#Proportion that are <0.05:
(sum(pvals<0.05))/length(pvals)
```

```
## [1] 0.04509218
```

```r
#Proportion that are <0.01:
(sum(pvals<0.01))/length(pvals)
```

```
## [1] 0.01021425
```

```
#Proportion that are <0.001:
(sum(pvals<0.001))/length(pvals)
```

```
## [1] 0.00124564
```

```
#In fact there are 5 less than 0.001:
sum(pvals<0.001)
```

```
## [1] 5
```

c. How many SNPs were tested for departures from Hardy-Weinberg equilibrium? Hint: How many P-values do you have? Second hint: Try using the "length" function. Save this value in the variable called "num_pval".

```
length(pvals)
```

```
## [1] 4014
```

```
num_pval <- length(pvals)
```

d. Say that you have "num_pval" total P-values. Assuming that the null hypothesis is true (i.e. all SNPs are in Hardy-Weinberg), the smallest P-values is expected to be 1/num_pval. The second smallest P-value is expected to be 2/num_pval. The third smallest P-value is expected to be 3/num_pval, etc. The largest P-value is expected to be num_pval/num_pval (or 1). Calculate the vector of expected P-values for the chi-square test. Save these in the vector called "exp_pvals".

```
exp_pvals <- 1:num_pval
exp_pvals <- exp_pvals/num_pval
max(exp_pvals)
```

```
## [1] 1
```

e. The observed P-values in the "pvals" vector are in the order that they SNPs appear across the chromosome. We need to sort them, smallest to largest. Use the "sort" function to sort the P-values. Store them in the vector "sort_pvals".

```
pvals <- sort(pvals)
```

f. In order to see what is happening with the small P-values (these are the ones we really care about), people often take the –log10(Pvalue). Find the –log10 of the observed and expected P-values. Store these in the vector "log_sort_pvals" and "log_exp_pvals".

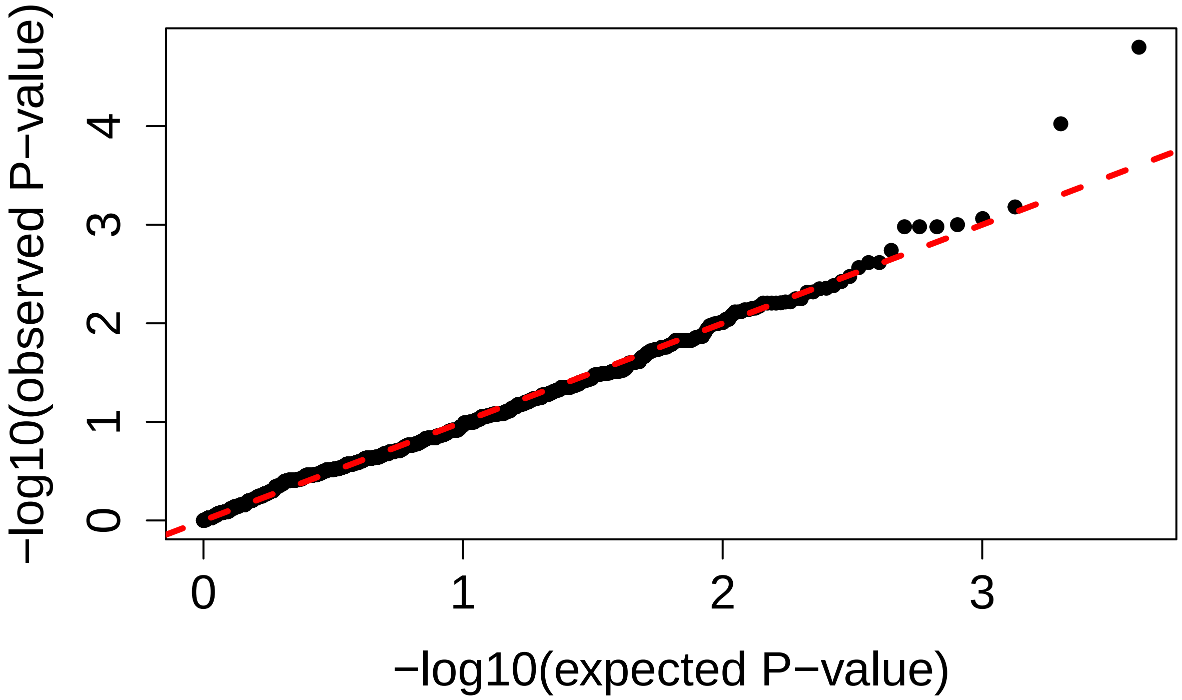
```
log_sort_pvals <- -log10(pvals)
log_exp_pvals <- -log10(exp_pvals)
```

g. You're ready to make the QQ plot! Plot the "log_sort_pvals" vs. the "log_exp_pvals".

```
#plot(log_exp_pvals, log_sort_pvals, xlab="-log10(expected P-value)",ylab="-log10(observed P-value)",ce
```

h. Where should these P-values fall under the null hypothesis? They should fall along the diagonal. Add a diagonal line to the QQ plot.

```
plot(log_exp_pvals, log_sort_pvals, xlab="-log10(expected P-value)",ylab="-log10(observed P-value)",cex
abline(a=0,b=1,col=2,lwd=3,lty=2)
```



2. Researchers are very interested in testing whether certain alleles are present in higher frequency in individuals with traits, such as type 2 diabetes. We have blood glucose levels for the 60 individuals in this study.

a. Load the file "pheno.sim.2014.txt". Store the phenotypes in a data frame called "zz". The second column in this file contains the blood glucose measurements. Hint: you probably want to use "header=T" in the "read.table" command.

```
zz <- read.table("phenotype-data.txt",header=T)
```

b. Find the value of the phenotype such that 25% of the individuals have a phenotype LESS than this value.

```
(quant.25 <- quantile(zz$glucose_mmolperL,probs=0.25))
```

```
##      25%
## 4.768756
```

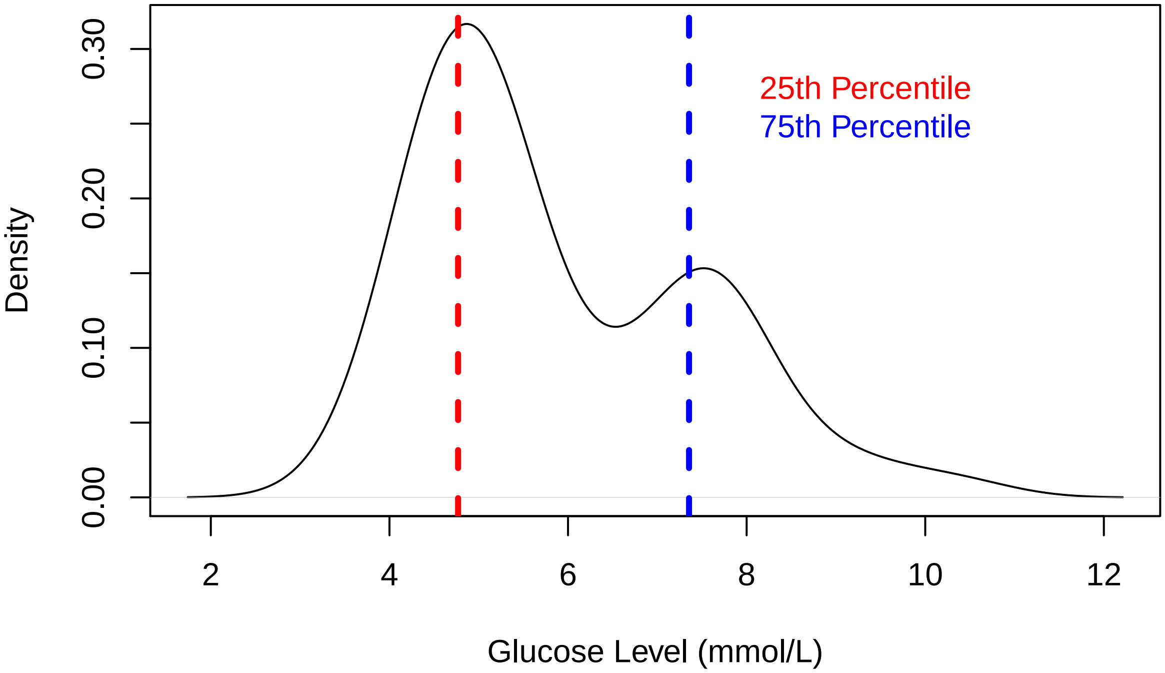
c. Find the value of the phenotype such that 25% of the individuals have a phenotype GREATER than this value (i.e. 75% of the individuals have a phenotype LESS than this value).

```
(quant.75 <- quantile(zz$glucose_mmolperL,probs=0.75))
```

```
##      75%
## 7.354975
```

d. Make a density plot of the distribution of phenotypes (i.e. the blood glucose levels). Add vertical lines to the plot to denote the 25% and 75% tails of the distribution.

```
plot(density(zz$glucose_mmolperL),main="",xlab="Glucose Level (mmol/L)")
abline(v=quant.25,col=2,lwd=3,lty=2)
abline(v=quant.75,col=4,lwd=3,lty=2)
legend(7.5,0.3,c("25th Percentile","75th Percentile"),text.col=c(2,4),bty='n')
```


Density
25th Percentile
75th Percentile
Glucose Level (mmol/L)