MED264: Statistics for Biomedical Informaticss

Jihoon Kim (j5kim@ucsd.edu)

11/7/2017

Topics covered

Complements MATH283 Statistical Methods in Bioinformatics (Glenn Tesler; http://www.math.ucsd.edu/~gptesler/283/calendar.html)

- Permutation test
- Sample size/statistical power
- Interrater agreement
- Logistic regression
- Survival analysis

Install R, RStudio and R packages

1. Download and install R, a language and environment for statistical computing and graphics (https://cran.r-project.org/)
2. Download and install RStudio Desktop (https://www.rstudio.com/products/rstudio/download/)
3. Install R packages

```
myPackageWishList = c( "CompQuadForm", "ggplot2", "irr", "knitr", "minqa",
                       "prLogistic", "pROC", "pwr", "ResourceSelection",  "survival")
for( i in myPackageWishList) {
  if (!is.element(i, installed.packages()[,1]))
     install.packages(i, dep = TRUE)
  require( i, character.only = TRUE, warn.conflicts = FALSE, quietly = TRUE )
}
```

```
## Warning: package 'lme4' was built under R version 3.4.2
```

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:boot':
##
##     melanoma
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:boot':
##
##     aml
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
## Type 'citation("pROC")' for a citation.
```

```
## ResourceSelection 0.3-2   2017-02-28
```

```
### For Mac OS. Install an R package directly from a URL to a tar file
if (!is.element("KATSP", installed.packages()[,1]))
    install.packages("http://www.biostat.umn.edu/~baolin/research/KATSP_0.1.0.tgz", repos=NULL)
require("KATSP",  character.only = TRUE,  warn.conflicts = FALSE, quietly = TRUE )
```

P-value

The p-value is the probability obtaining a test statistic as extreme as or more extreme than the observed statistic given that the null hypothesis is true

Understanding P-value through a permutation test

- A randomization test is a permutation test based on randomization (random assignment).
- Its goal is to test a null hypothesis about treatment effects in a randomized experiment.
- Can be used when the required distributional assumptions do NOT HOLD.

Permutation test algorithm

1. Compute an observed test statistic in the experiment data.
2. Permute data(=rearrange data) and compute a new test statistic. Repeat multiple times.
3. Calculate the proportion of permuted test statistic values greater or equal to the observed one.
4. This proportion is the P-value.

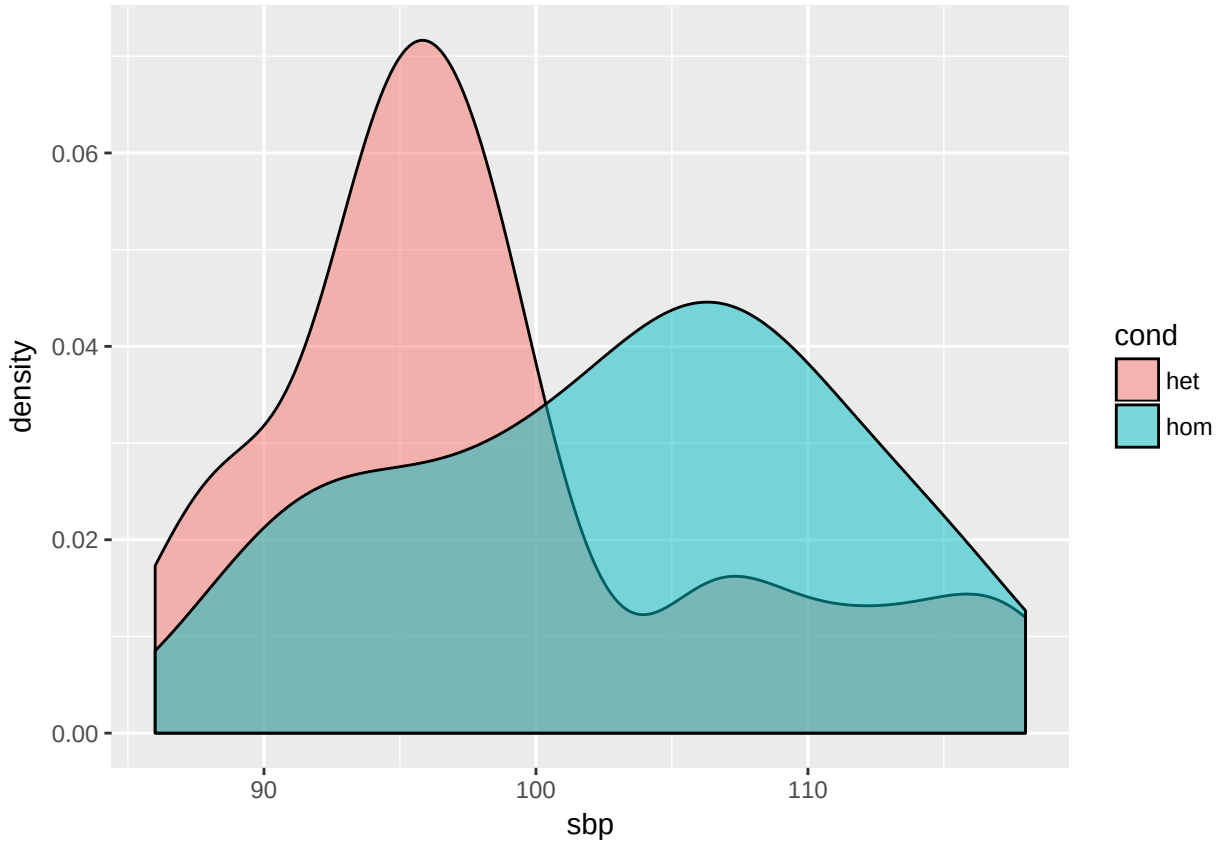
Permutation test example: Hypertension mouse model

- Systolic blood pressure was measured in 250 progeny from a backcross between two mouse strains.
- For simplicity, we focus on 50 (randomly chosen) mice genotyped at the D4Mit214 marker (although more markers were genotyped, chr4:45658442-45658566 bp, MGI:92846 http://www.informatics.jax.org/marker/MGI:92846).
- The question is to see if there is an association between the D4Mit214 marker genotype (binary) and blood pressure level (continuous).
- The values below show the systolic blood pressure (in mm of Hg) by the marker genotype, BA (heterozygous) or BB (homozygous) arranged in increasing order.
- Reference: Copeland et al. Science 1993. A genetic linkage map of the mouse: current applications and future prospects.

Plot of mouse hypertension data

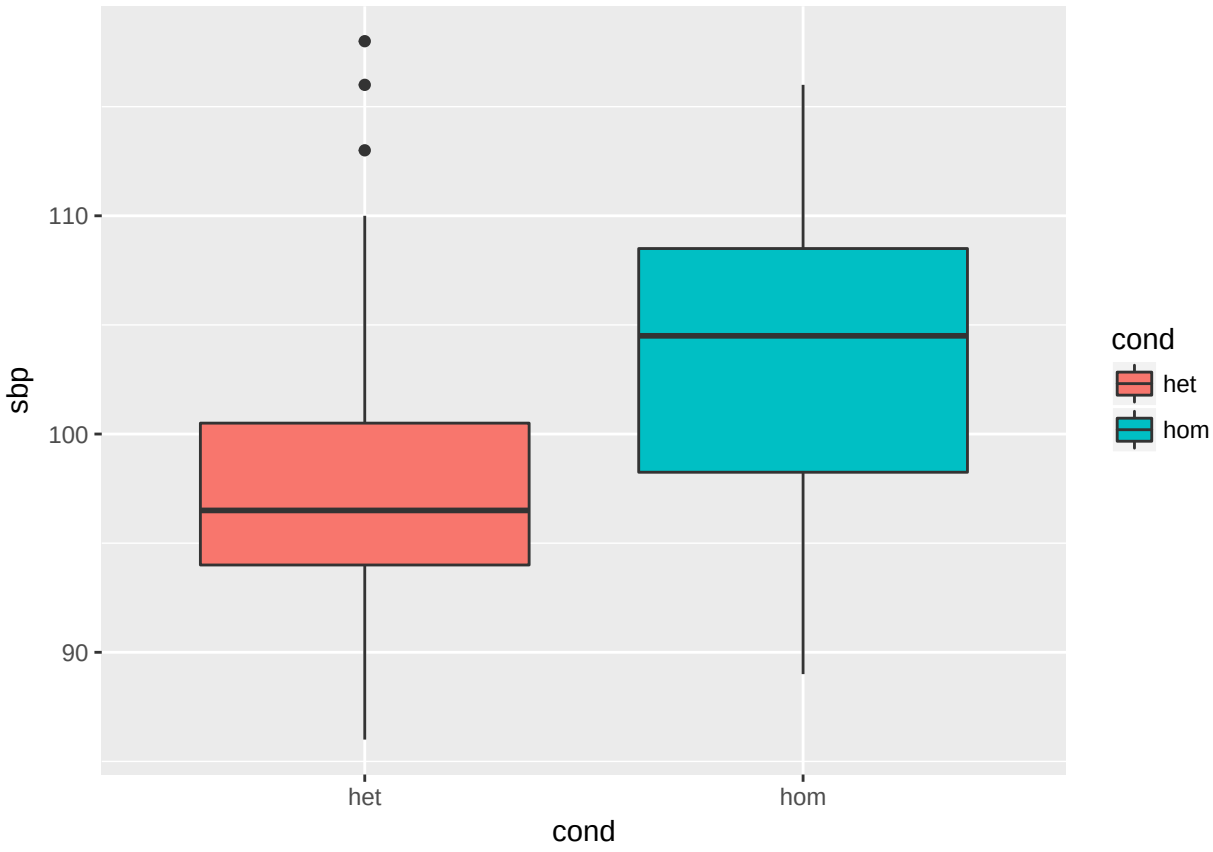
```
library(ggplot2)
het = c(86, 88, 89, 89, 92, 93, 94, 94, 94,
      95, 95, 96, 96, 97, 97, 98, 98, 99,
      99, 101, 106, 107, 110, 113, 116, 118)
hom = c(89, 90, 92, 93, 93, 96, 99, 99, 99, 102,
  103, 104, 105, 106, 106, 107, 108, 108, 110, 110,
  112, 114, 116, 116)
cond = c(rep("het", length(het)), rep("hom", length(hom)))
dat <- data.frame( cond, sbp=c(het, hom) )
```

```
ggplot(dat, aes(x = sbp, fill = cond)) +
   geom_density(alpha = 0.5)
```



Boxplot of hypertension data

```
library(ggplot2)
het = c(86, 88, 89, 89, 92, 93, 94, 94, 94,
      95, 95, 96, 96, 97, 97, 98, 98, 99,
      99, 101, 106, 107, 110, 113, 116, 118)
hom = c(89, 90, 92, 93, 93, 96, 99, 99, 99, 102,
  103, 104, 105, 106, 106, 107, 108, 108, 110, 110,
  112, 114, 116, 116)
cond = c(rep("het", length(het)), rep("hom", length(hom)))
dat <- data.frame( cond, sbp=c(het, hom) )
ggplot(dat, aes(x=cond, y=sbp, fill=cond, width=0.3)) +
  geom_boxplot()
```



R code for permutation test with hypertension data

```
het = c(86, 88, 89, 89, 92, 93, 94, 94, 94,
        95, 95, 96, 96, 97, 97, 98, 98, 99,
       99, 101, 106, 107, 110, 113, 116, 118)
hom = c(89, 90, 92, 93, 93, 96, 99, 99, 99, 102,
  103, 104, 105, 106, 106, 107, 108, 108, 110, 110,
  112, 114, 116, 116)
set.seed(2017)
diff.obs = mean(hom) - mean(het)
n.perm=5000
diff.perm=rep(NA, n.perm)
len.het=length(het)
len.hom=length(hom)
hethom = c(het, hom)
for (i in 1 : n.perm) {
    het.perm = sample(hethom, len.het, replace=TRUE)
    hom.perm = sample(hethom, len.hom, replace=TRUE)
    diff.perm[i] = mean(hom.perm) - mean(het.perm)
 }
p.perm = sum( abs(diff.perm) >= abs(diff.obs) ) / n.perm
print(p.perm)
```

```
## [1] 0.0454
```

How to calculate P-value

Recall the definition: the p-value is the probability obtaining a test statistic as extreme as or more extrement than the observed statistic given that the null hypothesis is true.

- In a parametric test (e.g. t-test), one makes assumptions about the parameters of the population distributions from which data are drawn and the p-value is calculated from this distribution function.
- In a non-parametric test (e.g. wilcox test), one makes no assumptions about the parameters of the population and the p-value is calculated based on ranks and probabilities.
- In a permutation test, there is no specific assumption about distributions, so test statistics are generated by permuting the data.

Statistical test of hypentension data

```
het = c(86, 88, 89, 89, 92, 93, 94, 94, 94,
      95, 95, 96, 96, 97, 97, 98, 98, 99,
      99, 101, 106, 107, 110, 113, 116, 118)
hom = c(89, 90, 92, 93, 93, 96, 99, 99, 99, 102,
  103, 104, 105, 106, 106, 107, 108, 108, 110, 110,
  112, 114, 116, 116)
t.test(het, hom)$p.value
```

```
## [1] 0.04824252
```

```
wilcox.test(het, hom, exact=FALSE)$p.value
```

```
## [1] 0.04417628
```

Statistical power

The power of a statistical test is the probability of correctly rejecting the null hypothesis when it is false and defined as 1 - beta = Pr(reject null hupothesis when it is false)

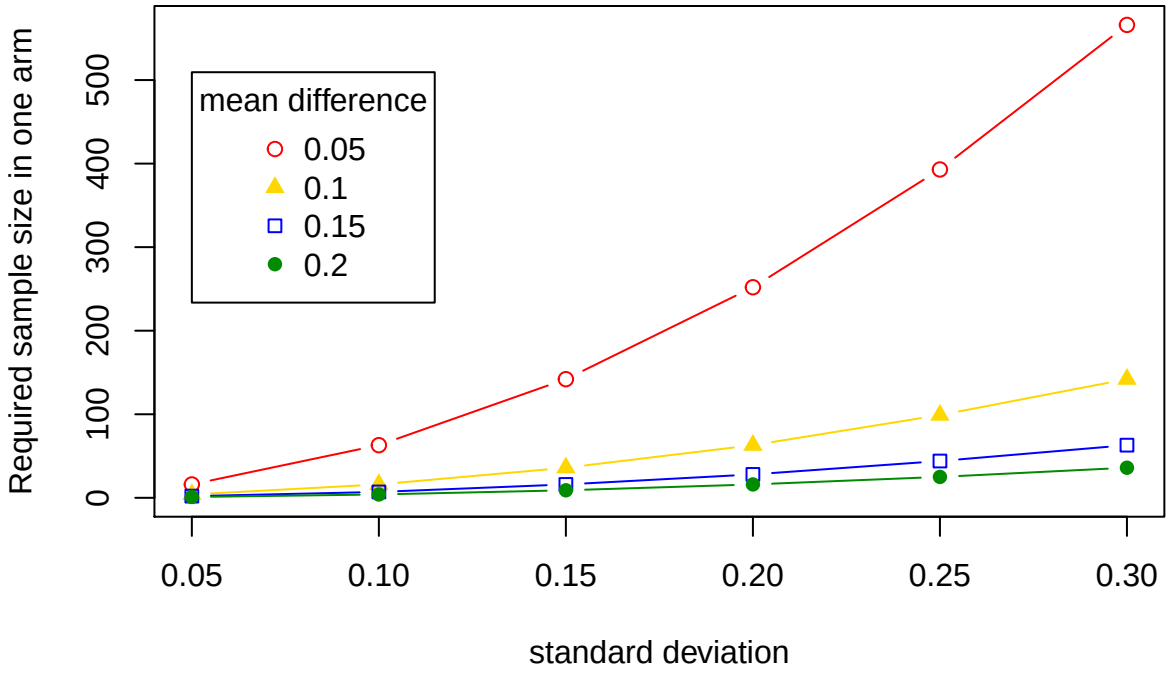
Example: Coronary heart disease

- Consider a clinical trial for evaluation of the effect of a test drug on cholesterol in patients with coronary heart disease (CHD).
- Cholesterol is the main lipid associated with atherosclerotic vascular disease (ASVD). The purpose of cholesterol testing is to identify patients at risk for atherosclerosis.
- About 75% of the cholesterol is bound to low density lipoproteins (LDLs) and 25% to high density lipoproteins (HDLs). Of these, LDL is the most directly associate with increased risk of CHD.
- A pharmaceutical company is interested in conducting a clinical trial to compare two cholesterol lowering agents for treatment of patients with CHD through a parallel design. The primary efficacy parameter is the LDL.
- Suppose a difference of 5% in percent change of LDL is considered of clinically meaningful difference.
- Assume that the standard deviation is 10%.
- What is the required sample size per group to achieve an 80% power at the 5% significance level?

- Reference: Chow,S et al. (2008) Sample size calculations in clinical research. Boca Raton,FL. Chapman and Hall/CRC.

Estimate required sample size in a study

```
## [1] 63
```



R code for sample size estimation about t-test

```r
# DEFINE A FUNCTION
ss.two_sample_mean = function(sigma, epsilon, alpha, beta) {
    term.top =  (qnorm(1-alpha/2) + qnorm(beta) )^2 * sigma^2 * 2
    term.bottom <-  abs(epsilon)^2
    return(  ceiling( term.top / term.bottom ) )
}

# RUN A FUNCTION FOR A CORONARY HEART DISEASE EXAMPLE
ss.two_sample_mean(0.1, 0.05, 0.05, 0.8)
# [1] 63

# VARY PARAMETERS
my.sigma = seq(0.05, 0.3, 0.05); my.epsilon = seq(0.05, 0.2, 0.05)
len.sigma = length(my.sigma); len.epsilon = length(my.epsilon);
my.ssize = matrix(1, nrow=len.sigma, ncol=len.epsilon, byrow=TRUE)
for(i in 1:len.sigma) {
   for(j in 1:len.epsilon) {
      my.ssize[i,j] = ss.two_sample_mean(my.sigma[i], my.epsilon[j], 0.05, 0.8)
    }
}
```

```
# DRAW A PLOT
my.color = c("red", "gold", "blue", "green4")
plot(my.sigma, my.ssize[,1], type="b", ylim=c(0, max(my.ssize)), col = my.color[1],
      xlab="standard deviation", ylab="Required sample size in one arm" )
points(my.sigma, my.ssize[,2], type="b", pch=17, col=my.color[2] )
points(my.sigma, my.ssize[,3], type="b", pch=22, col=my.color[3] )
points(my.sigma, my.ssize[,4], type="b", pch=16, col=my.color[4] )
legend(my.sigma[1], max(my.ssize)*0.9,  legend=my.epsilon, pch=c(1, 17, 22, 16),
    col = my.color, title="mean difference"  )
```

pwr, an R package for sample size estimation

```
library(pwr)
delta <- 0.05
sigma <- 0.1
d <- delta/sigma
pwr.t.test(d = d, sig.level = 0.05, power = 0.8, type = 'two.sample')$n
```

```
## [1] 63.76561
```

Advanced application: power calculation for variant-set based association tests

```
library(knitr)

power.commonvariants = matrix(NA, 5, 5)
power.commonvariants[,1] = c(1:5)*1000
power.commonvariants[,2] = c(2, 16, 52, 82, 96)
power.commonvariants[,3] = c(2, 16, 52, 82, 96)
power.commonvariants[,4] = c(2, 16, 51, 82, 96)
power.commonvariants[,5] = c(2, 20, 57, 86, 97)
colnames(power.commonvariants) = c("n", "MonteCarlo", "Davies", "WuPankow", "Lee" )
kable(power.commonvariants,
      caption="Estimated power for rare variants of
      G6PC2 gene at significance level 2.5 x 10^-6")
```

Table 1: Estimated power for rare variants of G6PC2 gene at significance level 2.5 x 10^-6

n	MonteCarlo	Davies	WuPankow	Lee
1000	2	2	2	2
2000	16	16	16	20
3000	52	52	51	57
4000	82	82	82	86
5000	96	96	96	97

```
power.rarevariants = matrix(NA, 5, 5)
power.rarevariants[,1] = c(4, 6, 8, 10, 12)*1000
power.rarevariants[,2] = c(7, 26, 54, 78, 92)
power.rarevariants[,3] = c(7, 26, 53, 78, 92)
power.rarevariants[,4] = c(7, 26, 53, 77, 91)
```

```
power.rarevariants[,5] = c(9, 30, 59, 81, 93)
colnames(power.rarevariants) = c("n", "MonteCarlo", "Davies", "WuPankow", "Lee" )
kable(power.rarevariants,
      caption="Estimated power for rare variants of
      G6PC2 gene at significance level 2.5 x 10^-6")
```

Table 2: Estimated power for rare variants of G6PC2 gene at significance level 2.5 x 10^-6

n	MonteCarlo	Davies	WuPankow	Lee
4000	7	7	7	9
6000	26	26	26	30
8000	54	53	53	59
10000	78	78	77	81
12000	92	92	91	93

Reference: Wuet al. Ann Hum Genet 2016, On Sample Size and Power Calculation for Variant Set-Based Association Tests (https://www.ncbi.nlm.nih.gov/pubmed/26831402)

R code: power calculation for variant-set based association tests

```
## load R packages
library(KATSP)
library(minqa)
library(CompQuadForm)

## set the seed for randomization
set.seed(2017)

## simulate genotype and es for QT
## simulate 1e4 by 20 genotypes from MVN with pairwise corr=0.1
Z1 = matrix(rnorm(1e4*20,0,sqrt(0.9)),1e4,20) + rnorm(1e4,0,sqrt(0.1))
Z2 = matrix(rnorm(1e4*20,0,sqrt(0.9)),1e4,20) + rnorm(1e4,0,sqrt(0.1))

## population MAF U[0.0005,0.02]
maf = runif(20, 0.0005,0.02)
q0 = qnorm(maf, lower=FALSE)
G = t( I(t(Z1)>q0) + I(t(Z2)>q0) )

## simulate es
VLD = cov(G)
MAF = colMeans(G)/2
Ves = runif(20,-0.25,0.25)
a1 = RVS.params(n=5e3,VLD,Ves,MAF)

## power
alpha = 2.5e-6
powerComparison =
  c( davies.pwr(alpha,a1$lambda,a1$delta),
     wu.pwr(alpha,a1$lambda,a1$delta),
     lee.pwr(alpha,a1$lambda,a1$delta),
```

```
     liu.pwr(alpha,a1$lambda,a1$delta) )
names(powerComparison) = c("Davies", "Wu", "Lee", "Liu")
powerComparison = round(powerComparison*100, 2)
powerComparison
```

```
## Davies     Wu    Lee    Liu
##  40.60  40.68  41.00  41.45
```

Reference: Wu et al. Ann Hum Genet 2016, On Sample Size and Power Calculation for Variant Set-Based Association Tests (https://www.ncbi.nlm.nih.gov/pubmed/26831402)

Interrater agreement

- inter-rater agreement, inter-rater reliability or concordance
- degree of agreement among raters
- Cohen's κ measures the inter-rater agreement between two raters
- κ values ranges between 0 and 1, 1 being the perfect agreemnt.
- Fleiss' κ measures the inter-rater agreement among more than two raters

Use of interrater agreement in biomedical science

- Ofori et al. J Clin Lipidol 2017, Comparison of 3 risk estimators to guide initiation of statin therapy for primary prevention of cardiovascular disease. (https://www.ncbi.nlm.nih.gov/pubmed/29050979)
- Girometti et al. Eur Radiol. 2017, Automated breast volume scanner (ABVS) in assessing breast cancer size: A comparison with conventional ultrasound and magnetic resonance imaging. (https://www.ncbi.nlm.nih.gov/pubmed/29018952)
- Grady et al. Pac Symp Biocomput. 2010, Finding unique filter sets in PLATO: a precursor to efficient interaction analysis in GWAS data. (https://www.ncbi.nlm.nih.gov/pubmed/19908384)
- Grelick et al. Acad Emerg Med. 2004, Performance of a novel clinical score, the Pediatric Asthma Severity Score (PASS), in the evaluation of acute asthma. (https://www.ncbi.nlm.nih.gov/pubmed/14709423)

Example: interrater agreement in CVD

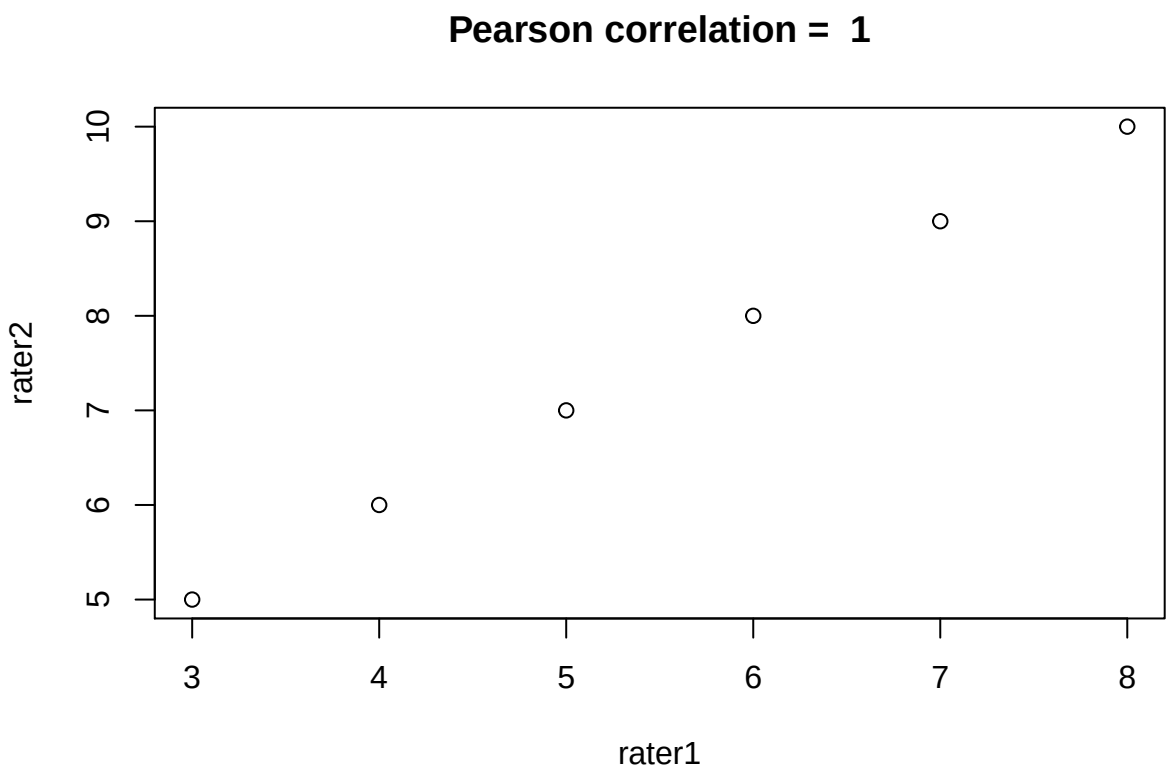
- TITLE: Comparison of 3 risk estimators to guide initiation of statin therapy for primary prevention of cardiovascular disease.
- AUTHOR: Ofori S, Dodiyi-Manuel S, Akpa MR.
- BACKGROUND: Among high-risk individuals, statins are beneficial for primary prevention of cardiovascular disease (CVD). In Nigeria, currently, there are no CVD prevention guidelines, so the use of CVD risk estimation to guide statin therapy is left to the discretion of the physician.
- OBJECTIVE: The objective of the study was to compare 3 CVD risk estimation tools in the evaluation of patients presenting to a tertiary hospital in Nigeria.
- METHODS: Cross-sectional study involving 295 patients with any CVD risk factors but not taking statins. Traditional CVD risk factors were assessed with a standard questionnaire and laboratory evaluation. Ten-year CVD risk was estimated with American College of Cardiology/American Heart Association Atherosclerotic Cardiovascular Disease (ACC/AHA ASCVD) Risk Estimator (2013), Framingham Risk Score (Framingham Risk Score [FRS] 2008), and the World Health Organisation/International

Society of Hypertension (WHO/ISH) risk prediction chart for Africa Region D. Kappa statistic was used to determine agreement among the estimators.

- RESULTS: The mean age was 48.4 ± 10.4 years; 60.7% were females. Risk factors for CVD were hypertension (56.3%), dyslipidemia (41.4%), diabetes (20%), obesity (28.5%), and cigarette smoking (4.4%). In all, 50.2%, 16.9%, and 15.2% were classified as high risk using the ACC/AHA ASCVD Risk Estimator, FRS 2008, and WHO/ISH risk chart, respectively. The agreement was moderate between FRS and WHO/ISH (Kappa 0.414, P < .001) and fair between ACC/AHA Estimator and WHO/ISH (Kappa 0.223, P < .001) and between ACC/AHA Estimator and FRS (Kappa 0.301, P < .001).
- REFERENCE: Ofori et al. J Clin Lipidol 2017, Comparison of 3 risk estimators to guide initiation of statin therapy for primary prevention of cardiovascular disease. (https://www.ncbi.nlm.nih.gov/pubmed/29050979)

Bias introduced if ratings were analyzed as a correlation

```
rater1 = c(3, 4, 5, 6, 7, 8)
rater2 = c(5, 6, 7, 8, 9, 10)

raters.m = cbind(rater1, rater2)

plot(rater1, rater2,
     main = paste("Pearson correlation = ", cor(rater1, rater2)) )
```



Reference: https://www.ncbi.nlm.nih.gov/pubmed/25852260

Data object for interrater agreement

```
library(irr)
data(diagnoses)
head(diagnoses)
```

```
##                    rater1                 rater2                 rater3
## 1             4. Neurosis            4. Neurosis            4. Neurosis
## 2 2. Personality Disorder 2. Personality Disorder 2. Personality Disorder
## 3 2. Personality Disorder       3. Schizophrenia       3. Schizophrenia
## 4                5. Other               5. Other               5. Other
## 5 2. Personality Disorder 2. Personality Disorder 2. Personality Disorder
## 6           1. Depression          1. Depression       3. Schizophrenia
##              rater4          rater5          rater6
## 1      4. Neurosis     4. Neurosis     4. Neurosis
## 2         5. Other        5. Other        5. Other
## 3 3. Schizophrenia 3. Schizophrenia        5. Other
## 4         5. Other        5. Other        5. Other
## 5      4. Neurosis     4. Neurosis     4. Neurosis
## 6 3. Schizophrenia 3. Schizophrenia 3. Schizophrenia
```

Cohen's Kappa for interrater agreement

```
library(irr)
data(diagnoses)
kappa2( diagnoses[,1:2] )
```

```
##  Cohen's Kappa for 2 Raters (Weights: unweighted)
##
##  Subjects = 30
##    Raters = 2
##     Kappa = 0.651
##
##         z = 7
##   p-value = 2.63e-12
```

Difference between logistic vs. linear regression

1. Outcome variable is binary (or dichotomous), not continous.
2. The nature of the relationship between the outcome and independent variables:

- linear regression:

$$E[Y|x] = \beta_0 + \beta_1 x$$

- logistic regression:

$$logit(E[Y|x]) = \beta_0 + \beta_1 x$$

where

$$logit(t) = log(\frac{t}{1-t})$$

- the conditional mean of the regression equation is formulated to be bounded between 0 and 1.

3. The conditional distribution of the outcome variable: binomial, not the normal.

Example: treatment for drug abuse

Goal was to compare treatment programs of different durations in the reduction of drug abuse and in the prevention of high-risk HIV behavior.

- DFREE: returned to drug use prior the study end date
- AGE: age at enrollment
- IVHX: IV drug use history at admission
- NRUGTX: number of prior drug treatment
- RACE: subject race (0 for white, and otherwise 1)
- TREAT: treatment site (0 for A and 1 for B)

Reference: Hosmer DW and Lemeshow S, Applied Logistic Regression, Johsn Wiley and Sons, Inc.

Estimated coefficients in a treatment study for drug abuse

```
library(pROC)
library(ResourceSelection)

rawdata = read.table("http://course1.winona.edu/bdeppa/Biostatistics/Data%20Sets/umaru.txt",
                 header = T)
uis = rawdata
uis$IVHX = factor(rawdata$IVHX)
uis$RACE = factor(rawdata$RACE)
uis$TREAT = factor(rawdata$TREAT)
uis$SITE = factor(rawdata$SITE)

m1 = glm(DFREE ~ AGE + NDRUGTX + IVHX  + RACE+ TREAT, data = uis, family="binomial")
round(summary(m1)$coef, 3)
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -2.356      0.550  -4.282    0.000
## AGE            0.051      0.017   2.944    0.003
## NDRUGTX       -0.063      0.026  -2.464    0.014
## IVHX2         -0.593      0.286  -2.070    0.038
## IVHX3         -0.760      0.249  -3.052    0.002
## RACE1          0.208      0.221   0.940    0.347
## TREAT1         0.439      0.199   2.204    0.028
```

```
### esimated coefficients and p-values
kable( round(summary(m1)$coef, 3) )
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.356	0.550	-4.282	0.000
AGE	0.051	0.017	2.944	0.003
NDRUGTX	-0.063	0.026	-2.464	0.014
IVHX2	-0.593	0.286	-2.070	0.038

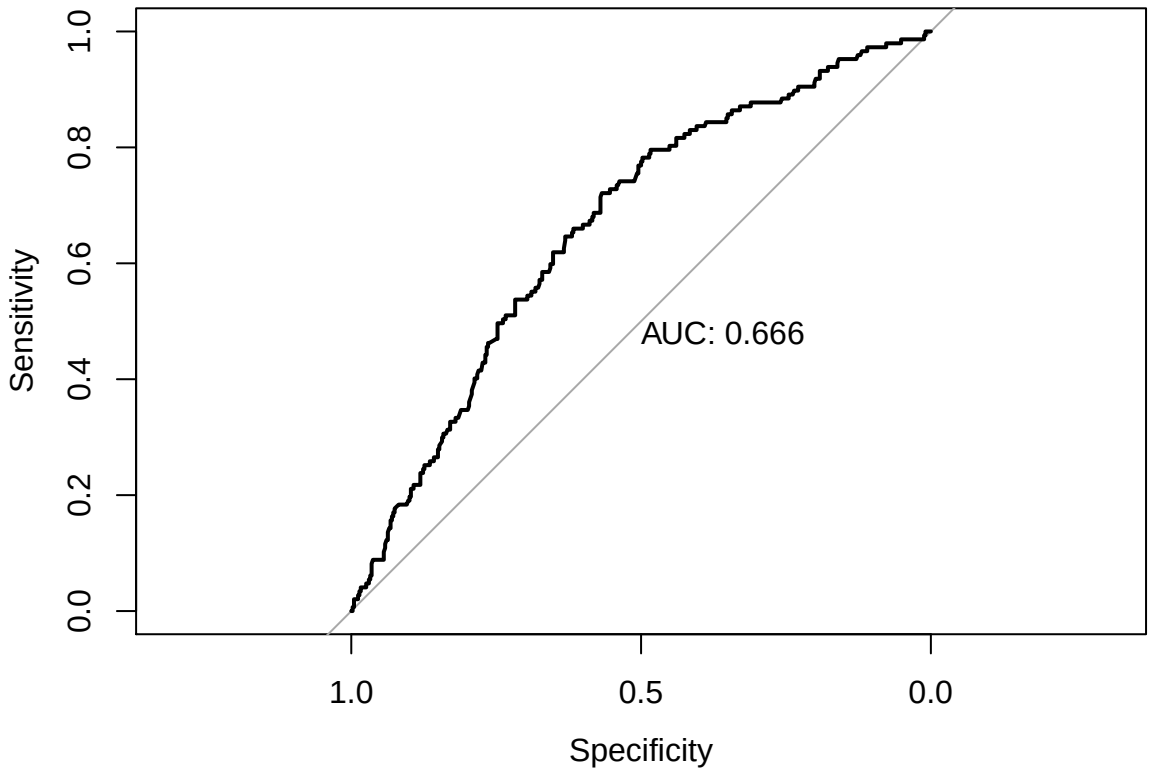
	Estimate	Std. Error	z value	Pr(>|z|)
IVHX3	-0.760	0.249	-3.052	0.002
RACE1	0.208	0.221	0.940	0.347
TREAT1	0.439	0.199	2.204	0.028

ROC Curve

```
### load R packages
library(pROC)
library(ResourceSelection)

### load data from URL
rawdata = read.table("http://course1.winona.edu/bdeppa/Biostatistics/Data%20Sets/umaru.txt",
                 header = T)

### convert to factor variable
uis = rawdata
uis$IVHX = factor(rawdata$IVHX)
uis$RACE = factor(rawdata$RACE)
uis$TREAT = factor(rawdata$TREAT)
uis$SITE = factor(rawdata$SITE)


### fit a logistic regression model
m1 = glm(DFREE ~ AGE + NDRUGTX + IVHX + RACE + TREAT, data = uis, family="binomial")

### Goodness of Fit test
hoslem.test(m1$y, m1$fitted)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  m1$y, m1$fitted
## X-squared = 5.5491, df = 8, p-value = 0.6976
```

```
### ROC curve and AUC
myroc = roc(m1$y, m1$fitted)
plot(myroc, print.auc=TRUE)
```



Survival Anslysis

- The most important difference between the outcome variables modeled via linear/logistic regression and survival analysis is the fact that we may observe the survival time partially
- For those subjects who died, it is the outcome variable of interest, the actual survival time.
- However, for subjects who were alive at the end of the study, or for subjects who were lost, the time variable indicates the length of follow-up, which is a partial or incomplete observation of survival time

These incomplete observations are referred to as being censored

- If we ignore the consoling and treat the censored observations as if they were measurements of survival time, then the resulting sample statistics are NOT estimators of the respective parameters of the survival time distribution.
- Reference: Hosmer,DW et al. (1999) Applied Survival Analysis, Danvers,MA. John Wiley and Sons, Inc.

Example: HIV

- A large HMO wishes to evaluate the survival time of its HIV positive member using a follow-up study
- Subjects were enrolled in the study from January 1, 1980 to December 3, 1991. The study ended December 31, 1995.
- After a confirmed diagnosis of HIV, members were followed until death due to AIDS or AIDS-related complications, until the end of the study or until the subject was lost to follow-up.
- We assume that there were no deaths due to other causes (e.g. auto accident). The primary outcome variable of interest is survival time after a confirmed diagnosis of HIV.

- Among 100 patients, 49 had a history of prior intravenous drug use.
- Is there a difference in the survival curves between IV drug users and non-IV drug users?

Data object for survival analysis

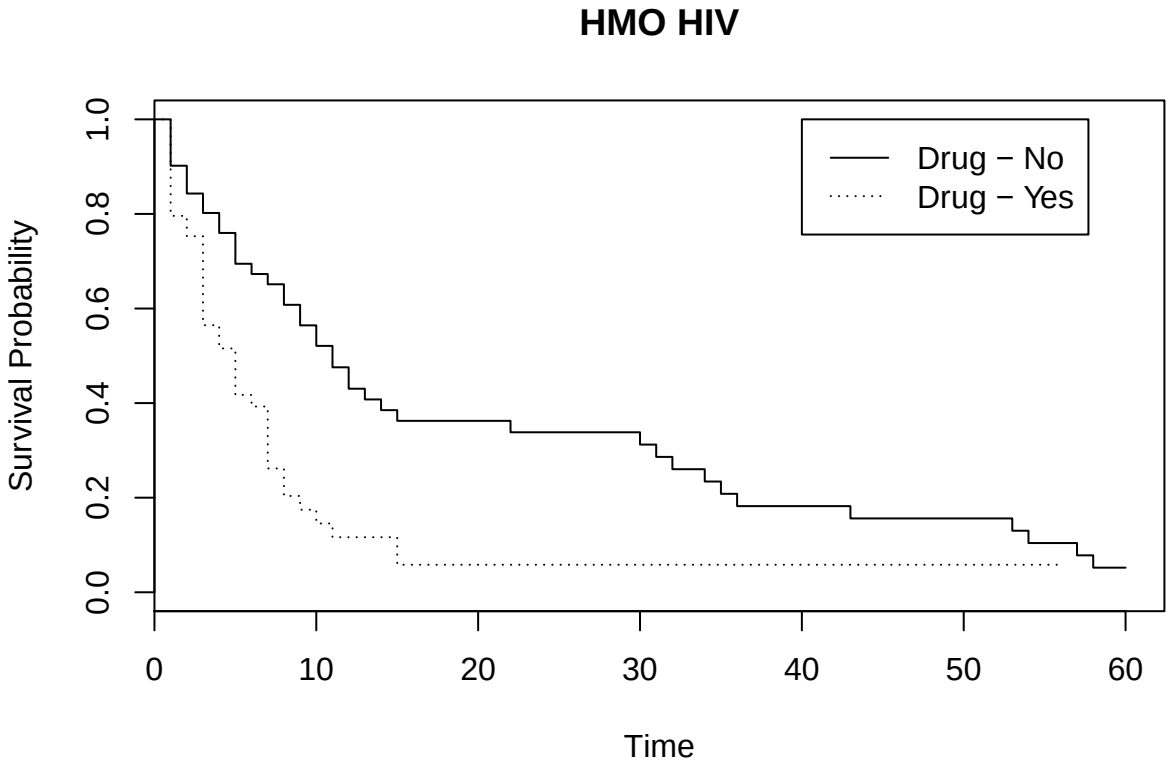
```
library(survival)
fname = "https://stats.idre.ucla.edu/stat/r/examples/asa/hmohiv.csv"
hmohiv = read.table(fname,sep=",", header = TRUE)
head(hmohiv)
```

```
##   ID time age drug censor    entdate     enddate
## 1  1    5  46    0      1 5/15/1990  10/14/1990
## 2  2    6  35    1      0 9/19/1989   3/20/1990
## 3  3    8  30    1      1 4/21/1991  12/20/1991
## 4  4    3  30    1      1  1/3/1991    4/4/1991
## 5  5   22  36    0      1 9/18/1989   7/19/1991
## 6  6    1  32    1      0 3/18/1991   4/17/1991
```

censor = 0 if the observation for survival time is complete and 1 if incomplete (=censored)

Survival curves

```
library(survival)
fname = "https://stats.idre.ucla.edu/stat/r/examples/asa/hmohiv.csv"
hmohiv = read.table(fname,sep=",", header = TRUE)
timestrata.surv <- survfit( Surv(time, censor) ~ strata(drug),
                            hmohiv,
                            conf.type="log-log")
plot(timestrata.surv, lty=c(1,3),
     main = "HMO HIV",
     xlab="Time", ylab="Survival Probability")
legend(40, 1.0, c("Drug - No", "Drug - Yes") , lty=c(1,3) )
```



Test for a difference between two survival curves

```
library(survival)
fname = "https://stats.idre.ucla.edu/stat/r/examples/asa/hmohiv.csv"
hmohiv = read.table(fname,sep=",", header = TRUE)
survdiff(Surv(time, censor) ~ drug, data=hmohiv, rho=0)
```

```
## Call:
## survdiff(formula = Surv(time, censor) ~ drug, data = hmohiv,
##     rho = 0)
##
##         N Observed Expected (O-E)^2/E (O-E)^2/V
## drug=0 51       42     54.9      3.02      11.9
## drug=1 49       38     25.1      6.60      11.9
##
##  Chisq= 11.9  on 1 degrees of freedom, p= 0.000575
```