# zadanie 1

#wykresy gęstości, dystrybuanty i funkcji przeżycia

#N(0,1), N(1,1), N(2,1)

curve(dnorm(x,0,1), from=-5, to=5, col="red",

main="gęstość rozkładu normalnego",

xlab="x", ylab="f(x)")

curve(dnorm(x,1,1), from=-5, to=5, col="green", add=T)

curve(dnorm(x,2,1), from=-5, to=5, col="blue", add=T)

legend("topleft", c("N(0,1)", "N(1,1)", "N(2,1)"), col

=c("red","green","blue"), lty=1);

curve(pnorm(x,0,1), from=-5, to=5, col="red",

main="dystrybuanta rozkładu normalnego",

xlab="x", ylab="F(x)")

curve(pnorm(x,1,1), from=-5, to=5, col="green", add=T)

curve(pnorm(x,2,1), from=-5, to=5, col="blue", add=T)

legend("topleft", c("N(0,1)", "N(1,1)", "N(2,1)"), col

=c("red","green","blue"), lty=1);

curve(1-pnorm(x,0,1), from=-5, to=5, col="red",

main="funkcja przeżycia rozkładu normalnego",

xlab="x", ylab="f(x)")

curve(1-pnorm(x,1,1), from=-5, to=5, col="green", add=T)

curve(1-pnorm(x,2,1), from=-5, to=5, col="blue", add=T)

legend("topright", c("N(0,1)", "N(1,1)", "N(2,1)"), col

=c("red","green","blue"), lty=1);

#N(0,1), N(0,0.5), N(0,2)

curve(dnorm(x,0,1), from=-5, to=5, col="red",

main="gęstość rozkładu normalnego",

xlab="x", ylab="f(x)")

curve(dnorm(x,0,0.5), from=-5, to=5, col="green", add=T)

curve(dnorm(x,0,2), from=-5, to=5, col="blue", add=T)

legend("topleft", c("N(0,1)", "N(0,0.5)", "N(0,2)"), col

=c("red","green","blue"), lty=1);

curve(pnorm(x,0,1), from=-5, to=5, col="red",

main="dystrybuanta rozkładu normalnego",

xlab="x", ylab="F(x)")

curve(pnorm(x,0,0.5), from=-5, to=5, col="green", add=T)

curve(pnorm(x,0,2), from=-5, to=5, col="blue", add=T)

legend("topleft", c("N(0,1)", "N(0,0.5)", "N(0,2)"), col

=c("red","green","blue"), lty=1);

curve(1-pnorm(x,0,1), from=-5, to=5, col="red",

main="funkcja przeżycia rozkładu normalnego",

xlab="x", ylab="f(x)")

curve(1-pnorm(x,0,0.5), from=-5, to=5, col="green", add=T)

curve(1-pnorm(x,0,2), from=-5, to=5, col="blue", add=T)

legend("topright", c("N(0,1)", "N(0,0.5)", "N(0,2)"), col

=c("red","green","blue"), lty=1);

# teraz za pomocą funkcji plot, a nie curve:

# do plota trzeba mieć zdefiniowane jakieś punkty

x <- seq(-5, 5, by=0.01)

plot(x, dnorm(x,0,1), type="l", col="red",

main="gęstość rozkładu normalnego N(0,1)",

xlab="x", ylab="f(x)")

# zadanie 2

# P(u-3b <= X <= u+3b) =~ 0,997

#dla N(0,1):

pnorm(3, 0, 1)-pnorm(-3, 0, 1) # zachodzi ;)

#dla rozkładu standaryzowanego:

x <- seq(-5,5,by=0.01)

y <- dnorm(x)

plot(x,y,type="l",main="reguła 3-sigmowa")

xx<-c(-3,x[x>=-3 & x<=3],3)

yy<-c(0,y[x>=-3 & x<=3],0)

polygon(xx,yy,col="yellow")

points(0,0,pch=16)

text(0,0.03,expression(paste(mu)))

arrows(-3,0.055,3,0.055,code=3)

text(0,0.08,expression(paste(mu,"+/-3",sigma)))

text(0,0.15,paste(round(100\*(pnorm(3)-pnorm(-3)),2),"%",sep=" "))

# zadanie 3

# N(173,6)

# a) P(X <= 179)

pnorm(179,173,6)

# b) P(167 < X < 180)

pnorm(180,173,6)-pnorm(167,173,6)

# c) P(X > 181) = 1 - P(X < 181)

1-pnorm(181,173,6)

pnorm(181, 173, 6, lower.tail=F) # równoważne

#d) P(X<x)<60 kwantyl rzędu 0.6

qnorm(0.6, 173, 6)

# zadanie 4

# wyznaczanie kwantyli

qnorm(0.95)

qnorm(0.975)

qt(0.95,10) #t-studenta o 10 stopniach swobody

qt(0.99,20)

qchisq(0.9,4) #chi kwadrat o czterech stopniach swobody

qchisq(0.95,10)

qf(0.95,2,10) #f-snedecora o (2,10) stopniach swobody

qf(0.99,3,18)

# zadanie 5

#wykresy gęstości rozkładów gamma

?dgamma

curve(dgamma(x,shape=1,scale=1),from=0,to=3,col="red",

main="gęstość rozkładu gamma",xlab="x", ylab="f(x)")

curve(dgamma(x,shape=0.5,scale=1),from=0,to=3,col="blue",add=T)

curve(dgamma(x,shape=2,scale=1),from=0,to=3,col="green",add=T)

curve(dgamma(x,shape=3,scale=1),from=0,to=3,col="black",add=T)

curve(dgamma(x,shape=2,scale=1),from=0,to=3,col="red",

main="gęstość rozkładu gamma", xlab="x", ylab="f(x)")

curve(dgamma(x,shape=2,scale=2),from=0,to=3,col="green",add=T)

curve(dgamma(x,shape=2,scale=3),from=0,to=3,col="blue",add=T)

# zadanie 6

# wykresy gęstości rozkładów chi-kwadrat

?dchisq

curve(dchisq(x,5),from=0,to=50,col="red",

main="gęstość rozkładu chi-kwadrat", xlab="x", ylab="f(x)")

curve(dchisq(x,10),from=0,to=50,col="green",add=T)

curve(dchisq(x,40),from=0,to=50,col="blue",add=T)

# zadanie 7

# wykresy gęstości rozkładów t-Studenta

?dt

curve(dt(x,1),from=0,to=4,col="red",

main="gęstość rozkładu t-studenta", xlab="x", ylab="f(x)")

curve(dt(x,5),from=0,to=4,col="blue",add=T)

curve(dt(x,30),from=0,to=4,col="green",add=T)

curve(dnorm(x),from=0,to=4,col="black",add=T)

# zadanie 8

# wykresy gęstości rozkładów F Snedecora

?df

curve(dt(x,10,5),from=0,to=20,col="red",

main="gęstość rozkładu F Snedecora", xlab="x", ylab="f(x)")

curve(dt(x,10,10),from=0,to=20,col="green",add=T)

curve(dt(x,10,20),from=0,to=20,col="blue",add=T)

curve(dt(x,5,2),from=0,to=5,col="red",

main="gęstość rozkładu F Snedecora", xlab="x", ylab="f(x)")

curve(dt(x,3,2),from=0,to=5,col="green",add=T)

curve(dt(x,2,2),from=0,to=5,col="blue",add=T)

curve(dt(x,2,1),from=0,to=22,col="red",

main="gęstość rozkładu F Snedecora", xlab="x", ylab="f(x)")

curve(dt(x,2,5),from=0,to=22,col="blue",add=T)

curve(dt(x,2,10),from=0,to=22,col="green",add=T)

curve(dt(x,2,20),from=0,to=22,col="black",add=T)

?dexp

curve(dexp(x),from=0,to=22,col="yellow",add=T)

#zadanie 9

# wykresy gęstości rozkładów beta

?dbeta

curve(dbeta(x,5,2),from=0,to=3,col="red",

main="gęstość rozkładu beta", xlab="x", ylab="f(x)")

curve(dbeta(x,2,2),from=0,to=3,col="blue",add=T)

curve(dbeta(x,1,1),from=0,to=3,col="green",add=T)

#zadanie 10

#rozkład mas prawdopodobieństwa dla bin

?dbinom

r1<-dbinom(0:10,10,0.5)

barplot(r1,names=0:10)

?barplot

r2<-dbinom(0:10,10,0.25)

barplot(r2,names=0:10)

r3<-dbinom(0:50,50,0.25)

barplot(r3,names=0:50)

#zadanie 11

?dgeom

#p=1/10

dgeom(1,1/10)

dgeom(2,1/10)

dgeom(3,1/10)

dgeom(4,1/10)

#P(X>11)

1-pgeom(11,1/10)

#zadanie 12

# n=200-5=195, k=10, m=5, x=0

?dhyper

dhyper(0,5,195,5)

#zadanie 13

#exp(0.0001)

#P(X>1000)

1-pexp(1000,0.0001)

1-pexp(10000,0.0001)

1-pexp(30000,0.0001)

qexp(0.1,0.0001)

#zadanie 14

#pois, EX=4

?dpois

#??????????????????????????????????????

#zadanie 15

# 0<x<1 0<y<x^2

n <- 10000

u <- runif(n)

v <- runif(n)

?runif

plot(u, v, xlim=c(0,1), ylim=c(0,1), pch='.') # punkty oznaczamy "kropka"

curve(x\*x, col="red", type="l", lwd=3, add=T)

z <- (v <= u\*u);

sum(z);

mean(z)

# x^2 < y < 1-x^2

n <- 10000

u <- runif(n)

v <- runif(n)

plot(u, v, xlim=c(0,1), ylim=c(0,1), pch='.') # punkty oznaczamy "kropka"

curve(x\*x, col="red", type="l", lwd=3, add=T)

curve(1-x\*x, col="blue", type="l", lwd=3, add=T)

w <- (v<= 1-u\*u);#??????????????????????????????????????

g <- (v>=u\*u)

sum(g)

sum(w);

sum(w)-sum(g)

mean(w)