FROM LAST TIME:

3. Let X represent the number of insurance policies sold by an agent in a day. The moment generating function of X is $M_X(t) = 0.45e^{t} + 0.35e^{2t} + 0.15e^{3t} + 0.05e^{4t}$, for $-\infty < t < \infty$. Calculate the standard deviation of X.

Differentiate: $M_X'(t) = 0.45e^{t} + 0.7e^{2t} + 0.45e^{3t} + 0.2e^{4t}$

Then: $E(X) = M_X'(0) = 0.45 + 0.7 + 0.45 + 0.2 = 1.8$

Differentiate Again: $M_X''(t) = 0.45e^{t} + 1.4e^{2t} + 1.35e^{3t} + 0.8e^{4t}$

Then: $E(X^2) = M_X''(0) = 0.45 + 1.4 + 1.35 + 0.8 = 4.0$

Thus: $Var(X) = E(X^2) - E(X)^2 = 4.0 - 1.8^2 = 0.76$ and $\sigma_X = \sqrt{0.76} = \boxed{0.87}$

Alternately, Recall: $M_X(t) = E(e^{tX}) = \sum_x e^{tx} P(X=x)$

x	1	2	3	4
$P(X=x)$	0.45	0.35	0.15	0.05

SIMULATING RANDOM VARIABLES

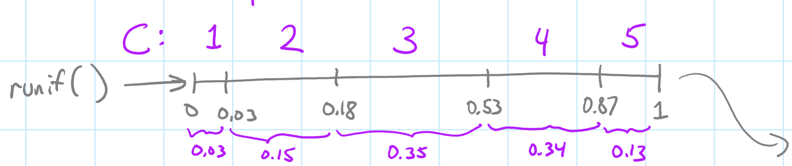
1. Suppose that C, the number of chips awarded in the game Plinko, has the following distribution:

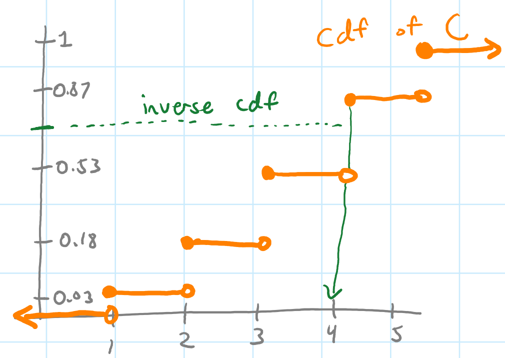
c	1	2	3	4	5
$p(c)$	.03	.15	.35	.34	.13

What are two ways of simulating values of C in **R**?

1. Inverse CDF Method

In R, runif() produces a random number uniformly between 0 and 1.






```
#simulate number of Plinko chips
chips <- NULL #list of results
for(i in 1:1000000){
  u <- runif(1)
  if(u < 0.03) {
    c <- 1 #one Plinko chip
  } else if(u < 0.18) {
```

```
      c <- 2
    } else if(u < 0.53) {
      c <- 3
    } else if(u < 0.87) {
      c <- 4
    } else {
      c <- 5
    }
    chips[i] <- c # store the result in our list
  }
  hist(chips)
  print(mean(chips))
  print(sd(chips))
```

2. Use the sample() function

sample(elements, sample size, replace = FALSE, probability vector)

For the number of chips:

sample($\underbrace{1:5}_{\text{list } 1,2,3,4,5}$, 1, replace = TRUE, c(0.03, 0.15, 0.35, 0.34, 0.13))

"combine"

Use simulation to estimate the mean and standard deviation of C.

Mean: about 3.39

Sd: about 0.98

2. Suppose that X, the winnings for one chip in Plinko, has the following distribution:

x	\$0	\$100	\$500	\$1000	\$10,000
$p(x)$	.39	.03	.11	.24	.23

Write a simulation of Plinko in R, taking into account both the number of chips a contestant earns and the amount of money won on each chip.

Plan:

repeat 10,000 times

- Simulate the number of chips, store in numchips
- Simulate numchips $\underbrace{\text{random variables with the distribution of } X}_{\text{winnings for each chip}}$
- add up the winnings

```
# simulate winnings in Plinko
winnings <- NULL
for(i in 1:10000){
```

```
  numchips <- sample(1:5, 1, replace=TRUE, c(.03, .15, .35, .34, .13))
  amounts <- sample(c(0, 100, 500, 1000, 10000), numchips, replace=TRUE, c(.39, .03, .11, .24, .23))
  winnings[i] <- sum(amounts) # total for this game, store in winnings list
}
hist(winnings)
```

What is the probability that a contestant wins more than $11,000?

```
sum(winnings > 11000)
mean(winnings > 11000)
```

SIMULATING COMMON DISTRIBUTIONS

rbinom(num. observations, n, p)

rpois(num. observations, λ)

rhyper(num. observations, M, N-M, n)

rgeom(num. observations, p)

3. Suppose that the number of customers buying flash drives in a store each week has a Poisson distribution with mean 80. Further suppose that the revenue per customer has the following distribution:

c	10	15	20	25	30
$p(x)$	.05	.10	.35	.40	.10

Use simulation to estimate the mean revenue per week. Then estimate the probability that the weekly revenue is at least $1800.

We didn't get to this in class, but here is some code for the simulation:

```
# number of customers who purchase flash drives (see Exercise 134)
rpois(1,80)

# simulate profit from 10000 customers
revenue <- NULL
vals <- seq(10,30,5)
probs <- c(.05,.10,.35,.40,.10)
for(i in 1:10000){
  num_custs <- rpois(1,80)
  rev <- sample(vals, num_custs, TRUE, probs)
  revenue[i]<-sum(rev)
}
mean(revenue)
sd(revenue)
mean(revenue >= 1800)
```