Ec/ACM/CS 112. Problem set 1. Template

Step 1: Programming stochastic simulations

1.A

```
library("ggplot2")
x1 <- rnorm(1000, mean=10, sd=10)
x2 <- rnorm(1000, mean=10, sd=10)
mean1 <- round(mean(x1), digits=2)
mean2 <- round(mean(x2), digits=2)

print(mean1)
```

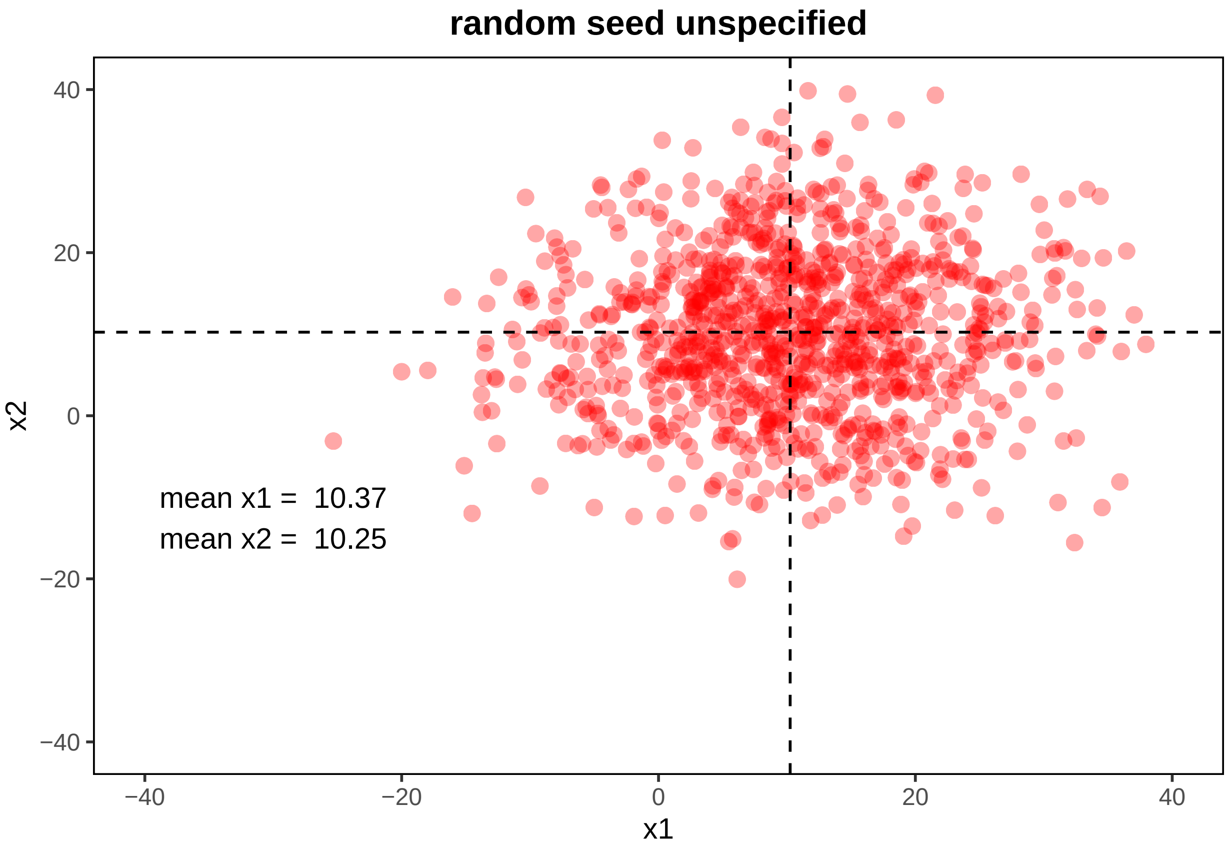
```
## [1] 10.37
```

```
print(mean2)
```

```
## [1] 10.25
```

```
df <- data.frame(xone = x1, xtwo = x2)
ggplot(df, aes(x=xone, y=xtwo)) + geom_point(size=3, color="red", alpha=0.35, stroke=0) +
  xlab("x1") + ylab("x2") + xlim(-40, 40) + ylim(-40, 40) +
  theme(panel.border = element_rect(size=0.5, fill=NA), panel.background = element_rect(fill="white"), p
  ggtitle("random seed unspecified") + geom_hline(yintercept=mean2, linetype="dashed") +
  geom_vline(xintercept=mean2, linetype="dashed") +
  annotate("text", x = -30, y = -10, label = paste("mean x1 = ", toString(mean1), sep=" ")) +
  annotate("text", x = -30, y = -15, label = paste("mean x2 = ", toString(mean2), sep=" "))
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



1.B

```
library("ggplot2")
set.seed(2021)
x1 <- rnorm(1000, mean=10, sd=10)
set.seed(2021)
x2 <- rnorm(1000, mean=10, sd=10)
mean1 <- round(mean(x1), digits=2)
mean2 <- round(mean(x2), digits=2)

print(mean1)
```

```
## [1] 10.13
```

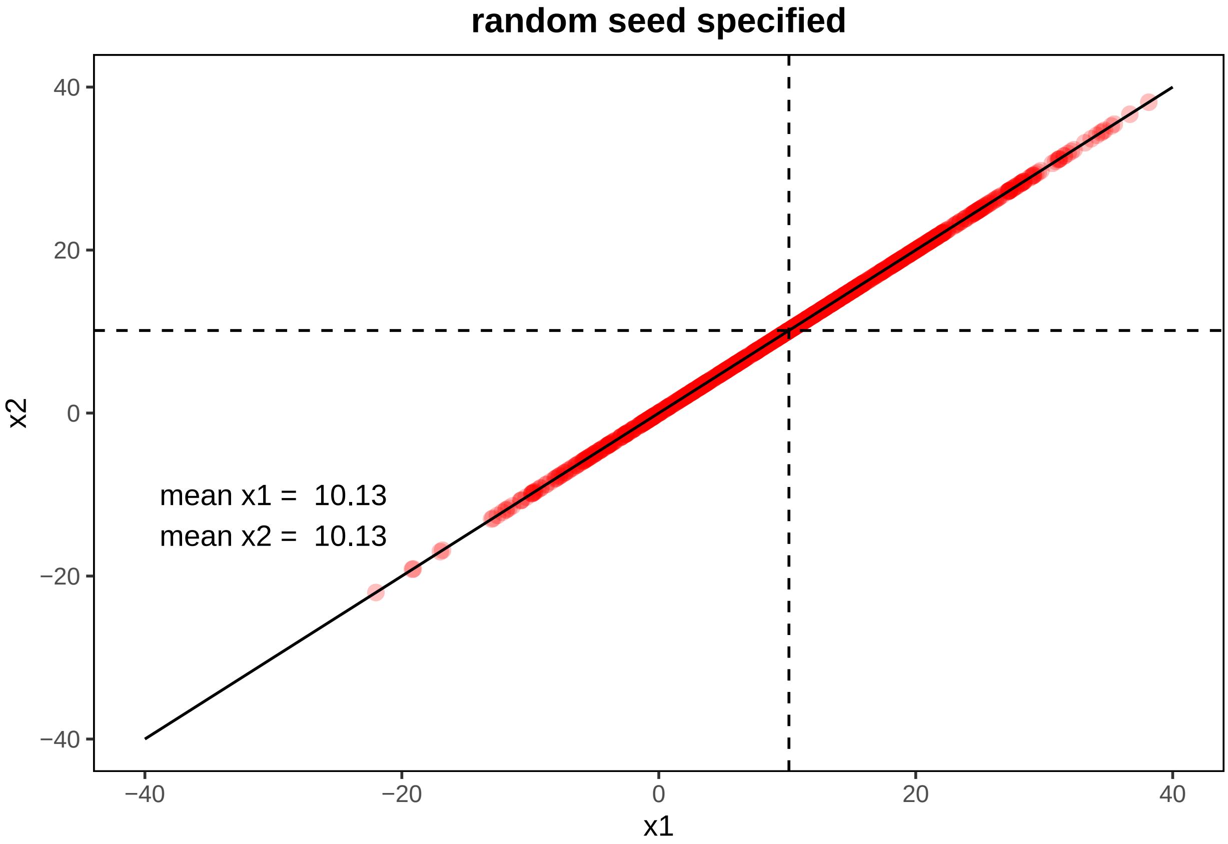
```
print(mean2)
```

```
## [1] 10.13
```

```
df <- data.frame(x = x1, y = x2)
df2 <- data.frame(x = -40:40, y=-40:40)

ggplot(df, aes(x, y)) + geom_point(size=3, color="red", alpha=0.25, stroke=0) +
  xlab("x1") + ylab("x2") + xlim(-40, 40) + ylim(-40, 40) +
  theme(panel.border = element_rect(size=0.5, fill=NA), panel.background = element_rect(fill="white"), p
  ggtitle("random seed specified") + geom_hline(yintercept=mean2, linetype="dashed") +
  geom_vline(xintercept=mean2, linetype="dashed") +
  annotate("text", x = -30, y = -10, label = paste("mean x1 = ", toString(mean1), sep=" ")) +
  annotate("text", x = -30, y = -15, label = paste("mean x2 = ", toString(mean2), sep=" ")) +
  geom_line(data = df2, color="black")
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



# Step 2: Simulating the hot-hand in basketball

2.A

```
library("ggplot2")
set.seed(2021)

simulate_player <- function(prob){
  vals = integer(25)
  for (i in 1:length(vals)){
    a <- runif(1)
    if (a <= prob){
      vals[i] <- 1
    }
  }
  vals
}

count_sequence <- function(vec){
  counter <- 1 #keeps track of index
  max_run <- 0 #holds value of longest run
  while (TRUE){
    if (counter > length(vec)){
      break
    }
    if (identical(vec[counter], 1)) {
      run_counter <- 1 #start a running tally for this sequence of 1's
      while (TRUE) {
        counter <- counter + 1 #jump to next index
        if (identical(vec[counter], 0) | (counter > length(vec))){
          ## we've reached the end of the run
          max_run <- max(max_run, run_counter)
          break
        } else {
          ## the run is not over
          run_counter <- run_counter + 1
        }
      }

    } else {
      ## this is not the beginning of a run
      ## jump to next index, return to top of while loop
      counter <- counter + 1
      ## check if we need to break the big while loop bc we've reached the last index value
      if (counter > length(vec)) {
        break
      }
    }
  }
  max_run
}
```

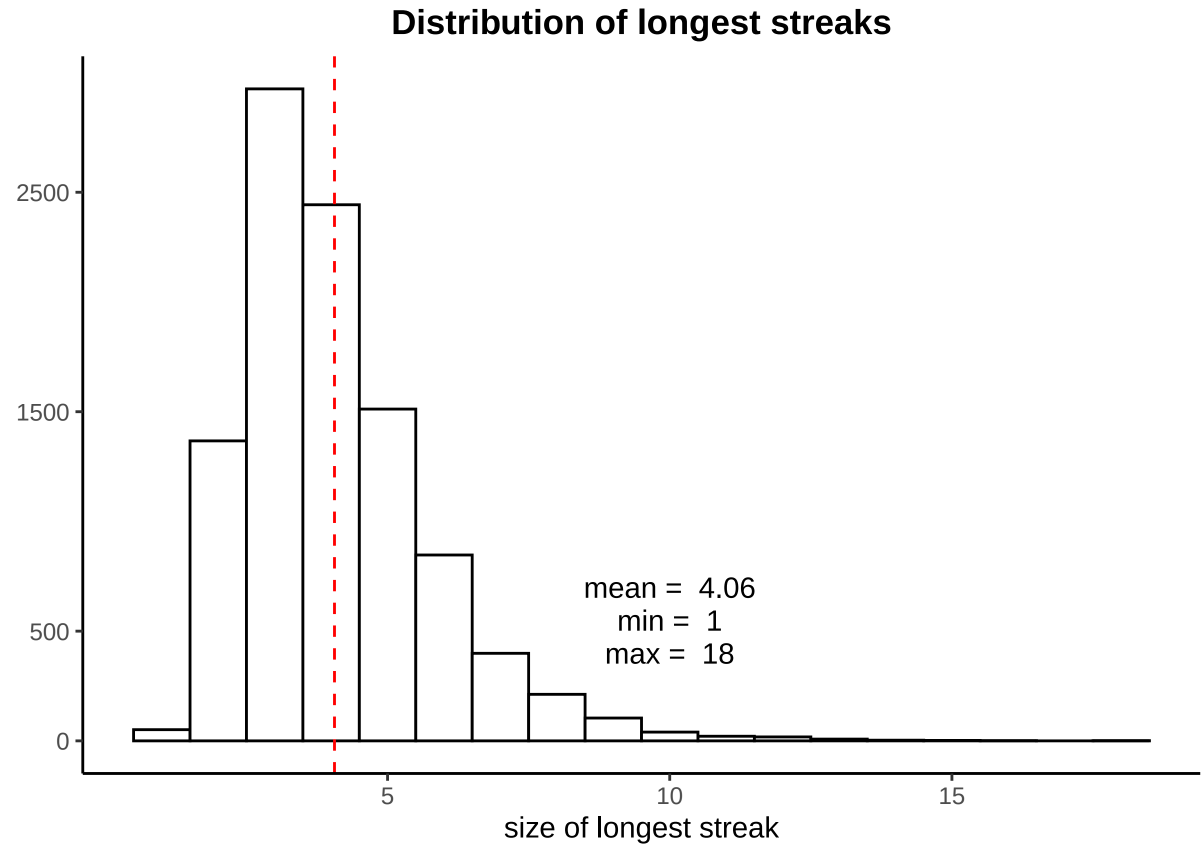
2.B

```
library("ggplot2")
sim_at_prob <- function(prob, players){
  data = integer(players)
  for (i in 1:players){
    values = simulate_player(prob)
    streak = count_sequence(values)
    data[i] <- streak
  }
  data
}

dat <- sim_at_prob(prob=0.5, players=10000)

mean <- round(mean(dat), digits=2)
df <- data.frame(x = dat)

ggplot(data=df, aes(x)) + geom_histogram(binwidth=1, color = "black", fill = "white") +
  xlab("size of longest streak") + ylab(" ") +
  theme( panel.background = element_rect(fill="white"), plot.title = element_text(hjust=0.5, color="bla
  ggtitle("Distribution of longest streaks") +
  geom_vline(xintercept=mean, linetype="dashed", color="red") +
  annotate("text", x = 10, y = 700, label = paste("mean = ", toString(mean), sep=" ")) +
  annotate("text", x = 10, y = 550, label = paste("min = ", toString(min(dat)), sep=" ")) +
  annotate("text", x = 10, y = 400, label = paste("max = ", toString(max(dat)), sep=" ")) +
  scale_x_continuous(breaks=c(5,10,15)) + scale_y_continuous(breaks=c(0,500,1500,2500))
```



2.C

```
library("ggplot2")
probs <- seq(0.1, 0.9, 0.05)
print(probs)
```

```
##  [1] 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70 0.75 0.80
## [16] 0.85 0.90
```

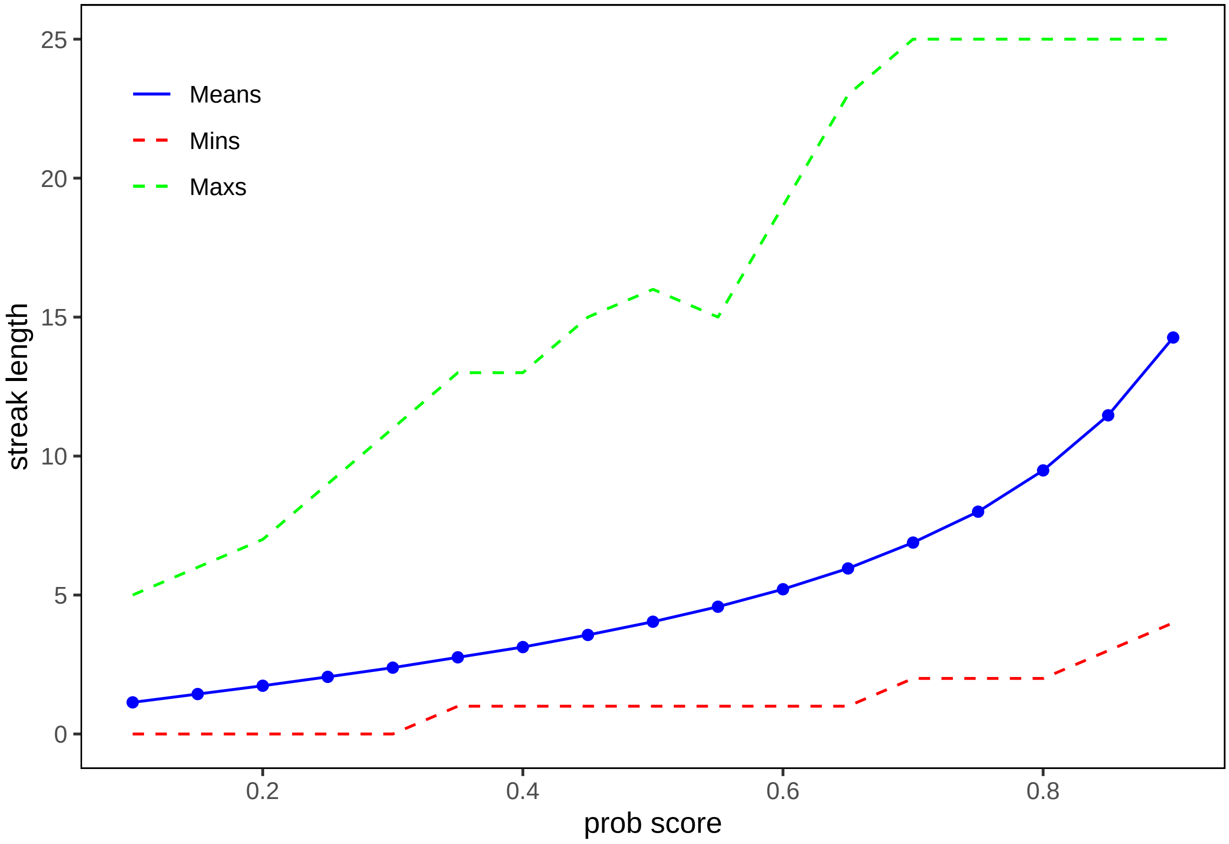
```
l = length(probs)
means <- double(l)
mins <- integer(l)
maxs <- integer(l)

for (i in 1:length(probs)){
  dat <- sim_at_prob(prob=probs[i], players=10000)
  means[i] <- mean(dat)
  mins[i] <- min(dat)
  maxs[i] <- max(dat)
}

df <- data.frame(p = probs, m1 = means, m2 = mins, m3 = maxs)
```

```
ggplot(data=df, aes(x=p)) + geom_point(aes(y = m1), color = "blue") + xlim(0, 1) +
  geom_line(aes(y = m1, colour = "Means", linetype="Means")) +
  geom_line(aes(y = m2, colour = "Mins", linetype = "Mins")) +
  geom_line(aes(y = m3, colour = "Maxs", linetype = "Maxs")) +
  scale_colour_manual(name=" ", breaks = c("Means", "Mins", "Maxs"), values = c("Means"= "blue", "Mins"=
  xlab("prob score") + ylab("streak length") +
  theme(legend.position = c(.1,.85), panel.border = element_rect(size=0.5, fill=NA), panel.background =
  labs(linetype = " ", color = " ") + scale_x_continuous(breaks=c(0, 0,.2, 0.4, 0.6, 0.8, 1.0)) +
  scale_linetype_manual( breaks = c("Means", "Mins", "Maxs"), values = c("Means"="solid", "Mins"="dashe
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```


Means
Mins
Maxs
streak length
prob score