Andrew ElHabr
ISyE 6832
2/4/18

Homework 1

1. (a) For all $\epsilon > 0$ and for all $m \geq 1$,

$$P(|X_n - 0| \leq \epsilon \ \forall n \geq m) = 1 - \frac{1}{n} + f(n, \epsilon),$$

in which $f(n,\epsilon) \in [0, \frac{1}{n}]$ and $\lim_{n\to\infty} f(n,\epsilon) = 0$. (This $f(n,\epsilon)$ term is only stated to be extremely pedantic. In reality, if $\epsilon \in (0,1)$, $f(n,\epsilon) = 0$ and the term is essentially unnecessary.) Therefore, this sequence converges to 0 almost surely because $1 - 1/n + f(n,\epsilon)$ tends to 1 as n approaches infinity. In addition, this sequence converges to 0 in L^1 because

$$\lim_{n\to\infty} E(|X_n - 0|) = \lim_{n\to\infty} (0(1 - \frac{1}{n}) + \sqrt{n}(\frac{1}{n})) = \lim_{n\to\infty} \frac{1}{\sqrt{n}} = 0.$$

However, this sequence does not converge to 0 in L^2 because

$$\lim_{n\to\infty} E(|X_n - 0|^2) = \lim_{n\to\infty} (0^2(1 - \frac{1}{n}) + \sqrt{n}^2(\frac{1}{n})) = \lim_{n\to\infty} 1 = 1 \neq 0.$$

(b) Since

$$\lim_{n\to\infty} P(X_n = 0) = \lim_{n\to\infty} (1 - (\frac{1}{2} + \frac{1}{n})) = \frac{1}{2}$$

and

$$\lim_{n\to\infty} P(X_n = 1) = \lim_{n\to\infty} (\frac{1}{2} + \frac{1}{n}) = \frac{1}{2}$$

is exactly equal to the distribution of $1_{[1/2,1]}(U)$, it must be that

$$X_n = 1_{[0,1/2+1/n)}(U) \implies 1_{[1/2,1]}(U).$$

However, for $\epsilon \in (0,1)$,

$$\lim_{n\to\infty} P(|X_n - 1_{[1/2,1]}(U)| \leq \epsilon) = \lim_{n\to\infty} P(\frac{1}{2} \leq U < \frac{1}{2} + \frac{1}{n}) = \lim_{n\to\infty} \frac{1}{n} = 0,$$

so $X_n = 1_{[0,1/2+1/n)}(U)$ does not converge in probability to $1_{[1/2,1]}(U)$.

(c) We know that for any $x \in [0,1]$,

$$P(X_n \leq x) = \sum_{k=0}^{\lfloor nx \rfloor} P(X = \frac{k}{n}) = \sum_{k=0}^{\lfloor nx \rfloor} \frac{1}{n+1} = \frac{\lfloor nx \rfloor + 1}{n+1}.$$

Now, since $(\lfloor nx \rfloor + 1) \in [nx, nx+1]$, it must be that

$$\lim_{n\to\infty} \frac{nx}{n+1} \leq \lim_{n\to\infty} \frac{\lfloor nx \rfloor + 1}{n+1} \leq \lim_{n\to\infty} \frac{nx+1}{n+1}.$$

Thus, since

$$\lim_{n\to\infty} \frac{nx}{n+1} = \lim_{n\to\infty} \frac{nx+1}{n+1} = x,$$

it must be that

$$\lim_{n\to\infty} \frac{\lfloor nx \rfloor + 1}{n+1} = x.$$

Therefore, $\lim_{n\to\infty} P(X_n \leq x) = x = P(U \leq x)$, i.e. $X_n \implies U$. (Also note that for both distributions, the cdfs are trivially 0 when $x \leq 0$ and 1 when $x \geq 1$.)

2. The cdf of $n(1-\max\{X_1, X_2, ..., X_n\})$ in which $X_1, X_2, ..., X_n$ are i.i.d. random variables with the uniform distribution on [0,1] is

$$\begin{aligned}
P(n(1-\max\{X_1,X_2,...,X_n\}) \le x) =& P(\max\{X_1,X_2,...,X_n\} \ge 1-\frac{x}{n}) \\
=& 1-P(\max\{X_1,X_2,...,X_n\} \le 1-\frac{x}{n}) \\
=& 1-P(X_1 \le 1-\frac{x}{n}, X_2 \le 1-\frac{x}{n}, ..., X_n \le 1-\frac{x}{n}) \\
=& 1-\prod_{i=1}^{n} P(X_i \le 1-\frac{x}{n}) \\
=& 1-(1-\frac{x}{n})^n.
\end{aligned}$$

Therefore,

$$\lim_{n\to\infty} P(n(1-\max\{X_1,X_2,...,X_n\}) \le x) = \lim_{n\to\infty} 1-(1-\frac{x}{n})^n = 1-e^{-x},$$

which is the cdf of an exponential distribution with $\lambda = 1$, i.e.

$$n(1-\max\{X_1,X_2,...,X_n\}) \implies \text{expo}(\lambda = 1).$$

3. Figure 1 shows the histograms for the project completion time for 10^3, 10^4, and 10^5 independent trials. The simulation was implemented in R.

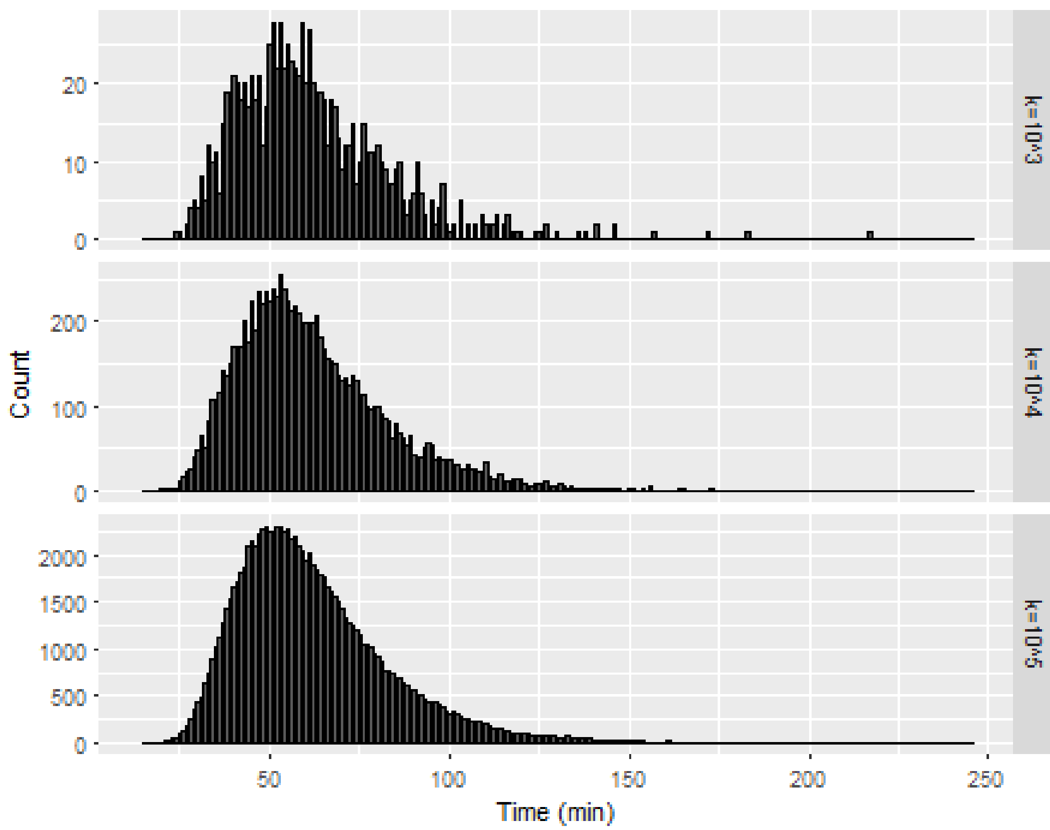

Figure 1: Project completion time histograms for k independent trials with bins for every 2 minutes.

Table 1 shows the mean project completion time and 95% CIs for 10^3, 10^4, and 10^5 independent trials. Tables 2-4 show the 5th, 25th, 50th, 75th and 95th percentiles of the project completion time and their corresponding 95% CIs for 10^3, 10^4, and 10^5 independent trials.

Table 1: Mean project completion time with 95% confidence intervals in minutes.

Trials	10^3	10^4	10^5
Mean Project Completion Time (min)	61.40 ± 1.33	61.89 ± 0.42	62.01 ± 0.13

Table 2: Project completion time percentiles with 95% confidence intervals in minutes for $k = 10^3$ trials.

Percentile	95% CI LB	Estimate	95% CI UB
5th	32.91	34.42	35.65
25th	45.20	46.55	48.16
50th	56.41	57.86	59.15
75th	69.42	72.21	74.11
90th	95.20	98.33	106.01

Table 3: Project completion time percentiles with 95% confidence intervals in minutes for $k = 10^4$ trials.

Percentile	95% CI LB	Estimate	95% CI UB
5th	34.30	34.64	35.03
25th	46.42	46.77	47.12
50th	57.24	57.72	58.18
75th	72.33	73.03	73.59
90th	101.71	103.33	104.85

Table 4: Project completion time percentiles with 95% confidence intervals in minutes for $k = 10^5$ trials.

Percentile	95% CI LB	Estimate	95% CI UB
5th	34.97	35.10	35.22
25th	46.73	46.85	46.97
50th	57.85	58.00	58.14
75th	72.54	72.76	72.97
90th	102.08	102.56	103.04

As the number of trials increases, the distribution of project completion time becomes "smoother" and looks closer to that of a right skewed normal distribution. Also, the CIs of the percentiles become much smaller with the number of trials as expected, but still overlap with the CIs corresponding to smaller amounts of trials.

4. (a) Let T_i be the time of the ith arrival and D_i be the time of the ith departure. Then, we have that

$$\begin{aligned} T_i &= \sum_{j=1}^{i} A_j, \\ D_i &= T_i + X_i + S_i. \end{aligned}$$

It is straightforward to see that we can now write the delay time of the ith entity explicitly as

$$X_i = \max\{D_{i-1}, T_i\} - T_i$$

for all $i = 2, 3, ...$ Therefore, for $i = 1, 2, ...,$

$$\begin{aligned} X_{i+1} &= \max\{D_i, T_{i+1}\} - T_{i+1} \\ &= \max\{D_i - T_{i+1}, 0\} \\ &= \max\{T_i + X_i + S_i - T_{i+1}, 0\} \\ &= \max\{X_i + S_i - A_{i+1}, 0\}. \end{aligned}$$

(b) We have that $\mathbb{E}[A] = \text{Var}(A) = 1$ and $\mathbb{E}[S] = \text{Var}(S) = 0.8$. Therefore,

$$\rho = \frac{\mathbb{E}[S]}{\mathbb{E}[A]} = \frac{0.8}{1} = 0.8, \quad c_A^2 = \frac{\text{Var}(A)}{(\mathbb{E}[A])^2} = \frac{1}{1^2} = 1, \quad c_S^2 = \frac{\text{Var}(S)}{(\mathbb{E}[S])^2} = \frac{0.8}{0.8^2} = 1.25.$$

Therefore,

$$\mu = \frac{\rho}{1-\rho} \times \frac{c_A^2 + c_S^2}{2} \times \mathbb{E}[S] = \frac{0.8}{1-0.8} \times \frac{1+1.25}{2} \times 0.8 = 3.6 \text{ min}.$$

(c) Table 5 shows the mean entity delay time and 95% CIs for 10^3, 10^4, and 10^5 consecutive entity delays for 100 independent trials each. Figure 2 shows that the 95% CI of the mean entity delay gets smaller as the number of consecutive entity delays increases and that each CI overlaps the steady-state mean $\mu = 3.6$ min. This figure also indicates that the transient phase of $\{X_i\}$ does not seem to matter since the 95% CIs are about where we expect to see them, especially for the experiment with the lowest number of consecutive entity delays. The simulation was implemented in R.

Table 5: Mean entity delay time with 95% confidence intervals in minutes.

Entity Delays	10^3	10^4	10^5
Mean Entity Delay Time (min)	3.54 ± 0.24	3.61 ± 0.07	3.60 ± 0.03

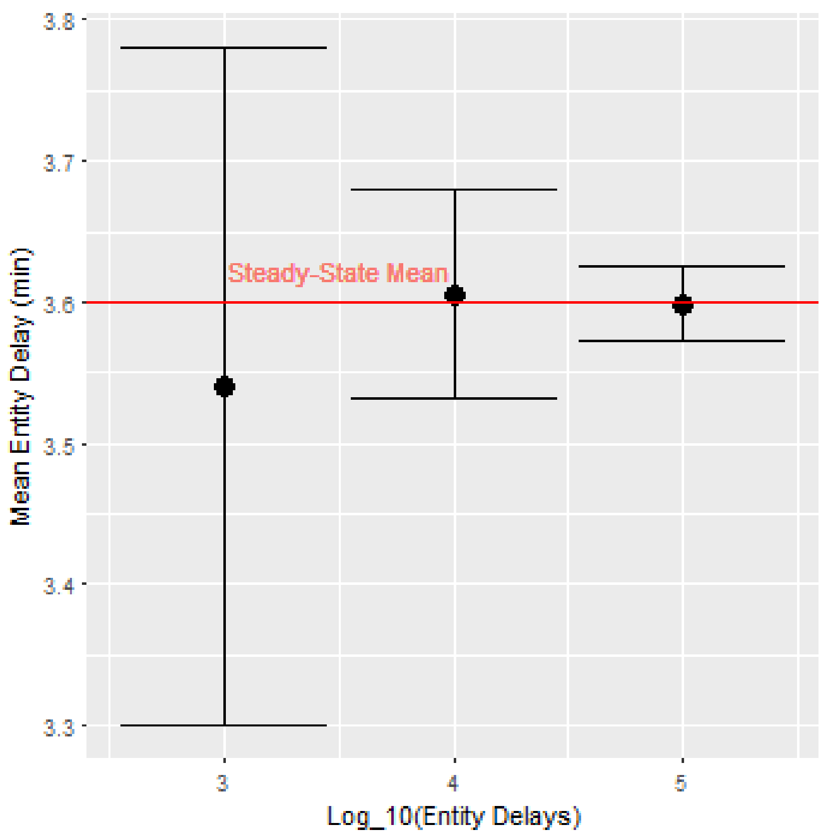

Figure 2: Mean entity delay times and their corresponding 95% confidence intervals for 10^3, 10^4, and 10^5 consecutive entity delays and 100 replications each.

(d) As depicted in Figure 3, only 15% of the 100 CIs contain the true mean $\mu = 3.6$ mins. This percentage may be surprising to some, but it should not be since these 95% CIs are for each replication and not for the mean delay time over all replications. We could expect that a 95% CI for the mean delay time over all replications would have a 95% chance of overlapping with the steady-state mean μ since the mean delay times for each replication are independent, but this same expectation should not be held for individual replications in which the delays are not independent from one another.

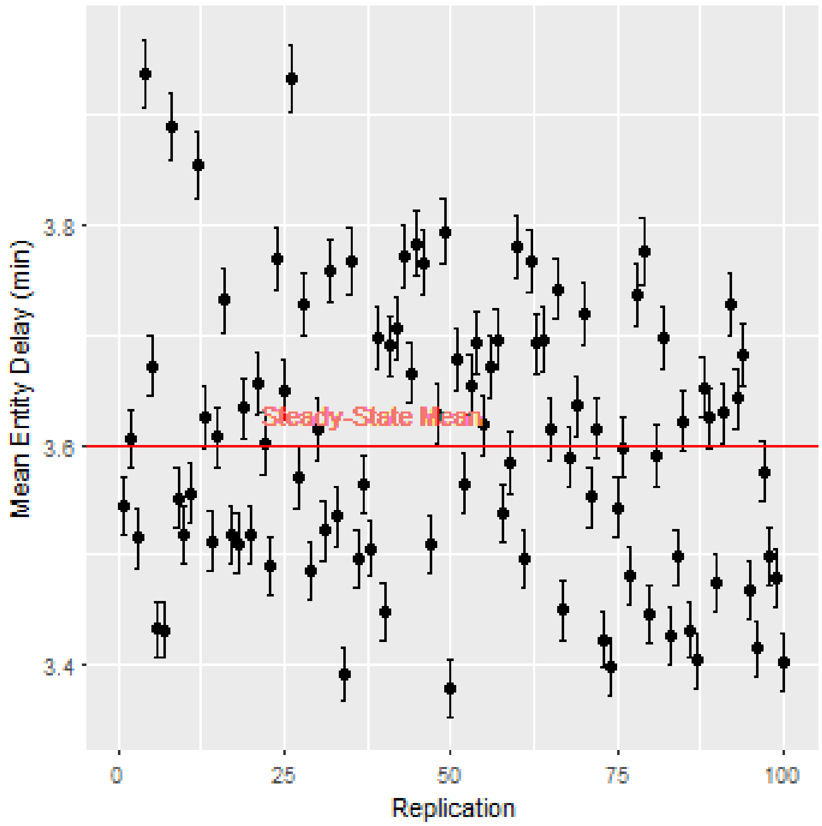

Figure 3: "Truncated" mean entity delay times and their corresponding 95% confidence intervals for each replication.

Appendix

```
##### Problem 3 #####

rm(list = ls())
library(ggplot2)

nodes <- 9
trials <- c(10^3,10^4,10^5)
arcs_norm <- c(13,5.5,5.2,3.2,3.2)
arcs_expo <- c(7,16.5,14.7,6,10.3,4,20,16.5)

quantiles <- c(0.05,0.25,0.5,0.75,0.95)
mean_completion_times <- rep(0,length(trials))
mean_completion_times_sd <- rep(0,length(trials))
mean_completion_times_95CI_HL <- rep(0,length(trials))
percentile_estimates <- array(0,c(length(trials),length(quantiles)))
percentile_estimates_95CI_LB <- array(0,c(length(trials),length(quantiles)))
percentile_estimates_95CI_UB <- array(0,c(length(trials),length(quantiles)))
durations_per_trial <- list()

set.seed(42)
for (k in 1:length(trials)){
  arc_lengths <- array(0,c(trials[k],length(arcs_norm)+length(arcs_expo)))
  durations <- array(0,c(trials[k],nodes))
  for (i in 1:trials[k]){
    arcs_norm_sim <- pmax(rnorm(length(arcs_norm),arcs_norm,arcs_norm/4),0)
    arcs_expo_sim <- rexp(length(arcs_expo),1./arcs_expo)
    arc_lengths[i,] <- c(arcs_norm_sim,arcs_expo_sim)
    durations[i,2] <- arc_lengths[i,1]
    durations[i,3] <- max(arc_lengths[i,2],durations[i,2]+arc_lengths[i,6])
    durations[i,4] <- durations[i,2]+arc_lengths[i,3]
    durations[i,5] <- durations[i,2]+arc_lengths[i,9]
    durations[i,6] <- max(durations[i,2]+arc_lengths[i,7],durations[i,3]+arc_lengths[i,8],durations[i,5]+
        arc_lengths[i,12])
    durations[i,7] <- durations[i,4]+arc_lengths[i,10]
    durations[i,8] <- max(durations[i,5]+arc_lengths[i,11],durations[i,7]+arc_lengths[i,5])
    durations[i,9] <- max(durations[i,6]+arc_lengths[i,4],durations[i,8]+arc_lengths[i,13])
  }

  mean_completion_times[k] <- mean(durations[,9])
  mean_completion_times_sd[k] <- sd(durations[,9])
  mean_completion_times_95CI_HL[k] <- qt(0.975,trials[k]-1)*mean_completion_times_sd[k]/sqrt(trials[k])
  percentile_estimates[k,] <- unname(quantile(durations[,9],quantiles))
  percentile_estimates_95CI_LB[k,] <- unname(quantile(durations[,9],quantiles-qnorm(0.975)*sqrt(quantiles
      *(1-quantiles)/trials[k])))
  percentile_estimates_95CI_UB[k,] <- unname(quantile(durations[,9],quantiles+qnorm(0.975)*sqrt(quantiles
      *(1-quantiles)/trials[k])))

  durations_df <- data.frame(trial = k, durations = durations[,9])
  durations_per_trial[[k]] <- durations_df
  }

mean_completion_times
mean_completion_times_95CI_HL
percentile_estimates
percentile_estimates_95CI_UB
percentile_estimates_95CI_LB

durations_to_plot <- do.call(rbind,durations_per_trial)
trial_names <- list("1"="k=10^3","2"="k=10^4","3"="k=10^5")
trial_labeller <- function(variable,value){
  return(trial_names[value])
}
png("ISyE_6832_Homework_1_Problem_3.png", width = 450, height = 350)
suppressWarnings(ggplot(durations_to_plot, aes(x=durations)) +
  geom_histogram(binwidth=1,color="black") +
  facet_grid(trial~.,scales="free",labeller=trial_labeller) +
  labs(x="Project Completion Time (min)",y = "Count"))
dev.off()



##### Problem 4c #####

rm(list = ls())

replications <- 100
entities <- c(10^3,10^4,10^5)
```

```r

Y_bar <- rep(0,length(entities))
Y_bar_95CI_UB <- rep(0,length(entities))
Y_bar_95CI_LB <- rep(0,length(entities))
Y_bar_95CI_HL <- rep(0,length(entities))

set.seed(42)
for(k in 1:length(entities)){
  arrivals <- array(0,c(entities[k],replications))
  services <- array(0,c(entities[k],replications))
  delays <- array(0,c(entities[k],replications))
  for(j in 1:replications){
    arrivals[-1,j] <- rexp(entities[k]-1,1)
    services[,j] <- rgamma(entities[k],0.8,1)
    for(i in 2:entities[k]){
      delays[i,j] <- max(delays[i-1,j]+services[i-1,j]-arrivals[i,j],0)
    }
  }
  delay_means <- colMeans(delays)
  Y_bar[k] <- mean(delay_means)
  Y_bar_95CI_UB[k] <- Y_bar[k] + qt(0.975,replications-1)*sd(delay_means)/sqrt(replications)
  Y_bar_95CI_LB[k] <- Y_bar[k] - qt(0.975,replications-1)*sd(delay_means)/sqrt(replications)
  Y_bar_95CI_HL[k] <- qt(0.975,replications-1)*sd(delay_means)/sqrt(replications)
}

Y_bar
Y_bar_95CI_HL

Y_df <- data.frame(log10_entities = log10(entities),
                   mean = Y_bar,
                   LB = Y_bar_95CI_LB,
                   UB = Y_bar_95CI_UB)
steady_state_mean <- 3.6

png("ISyE_6832_Homework_1_Problem_4c.png", width = 350, height = 350)
ggplot(Y_df, aes(x = log10_entities, y = mean)) +
  geom_point(size = 4) +
  geom_errorbar(aes(ymax = UB, ymin = LB)) +
  geom_hline(yintercept= steady_state_mean, color="red") +
  geom_text(aes(log10(entities)[1]+0.5, steady_state_mean, label = "Steady-State Mean", color="red",
      vjust = -1)) +
  theme(legend.position="none") +
  labs(x = "Log_10(Entity Delays)", y = "Mean Entity Delay (min)")
dev.off()



##### Problem 4d #####

rm(list = ls())
library(matrixStats)

replications <- 100
entities <- 10^5
extra <- 1000

set.seed(42)
for(k in 1:length(entities)){
  arrivals <- array(0,c(entities[k]+extra,replications))
  services <- array(0,c(entities[k]+extra,replications))
  delays <- array(0,c(entities[k]+extra,replications))
  for(j in 1:replications){
    arrivals[-1,j] <- rexp(entities[k]+extra-1,1)
    services[,j] <- rgamma(entities[k]+extra,0.8,1)
    for(i in 2:entities[k]+extra){
      delays[i,j] <- max(delays[i-1,j]+services[i-1,j]-arrivals[i,j],0)
    }
  }
  X_bar <- colMeans(delays[(1+extra):(entities[k]+extra),])
  X_bar_sd <- colSds(delays[(1+extra):(entities[k]+extra),])
  X_bar_95CI_UB <- X_bar + qnorm(0.975,0,1)*X_bar_sd/sqrt(entities[k])
  X_bar_95CI_LB <- X_bar - qnorm(0.975,0,1)*X_bar_sd/sqrt(entities[k])
  X_bar_95CI_HL <- qnorm(0.975,0,1)*X_bar_sd/sqrt(entities[k])
}

X_bar
X_bar_95CI_HL
steady_state_mean <- 3.6
contain_true_mean <- sum(X_bar_95CI_LB <= steady_state_mean & X_bar_95CI_UB >= steady_state_mean)/
    replications
contain_true_mean
```

```
X_df <- data.frame(replication = c(1:replications),
                   mean = X_bar,
                   LB = X_bar_95CI_LB,
                   UB = X_bar_95CI_UB)

png("ISyE_6832_Homework_1_Problem_4d.png", width = 350, height = 350)
ggplot(X_df, aes(x = replication, y = mean)) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymax = UB, ymin = LB)) +
  geom_hline(yintercept= steady_state_mean, color="red") +
  geom_text(aes(38.5, steady_state_mean, label = "Steady-State Mean", color="red", vjust = -1)) +
  theme(legend.position="none") +
  labs(x = "Replication", y = "Mean Entity Delay (min)")
dev.off()
```