Assignment 11

Due Friday, November 22, 11:59 PM CT

Kyle Yeo

```r
vb_team <- read_csv("C:/stat_240/data/volleyball-team-2019.csv")
vb_match <- read_csv("C:/stat_240/data/vb-division1-2019-all-matches-corrected.csv") %>%
  mutate(index = row_number()) %>%
  select(index,everything())
```

Problems

1

Multiple choice: The least-squares regression line is

(a) the line that makes the sum of the squares of the vertical distances of the data points to the line as small as possible

(b) the line that best splits the data in half, with half of the points above the line and half below the line

(c) the line that makes the correlation of the data as large as possible.

(d) all of the above

(e) a and b

(f) a and c

(g) b and c

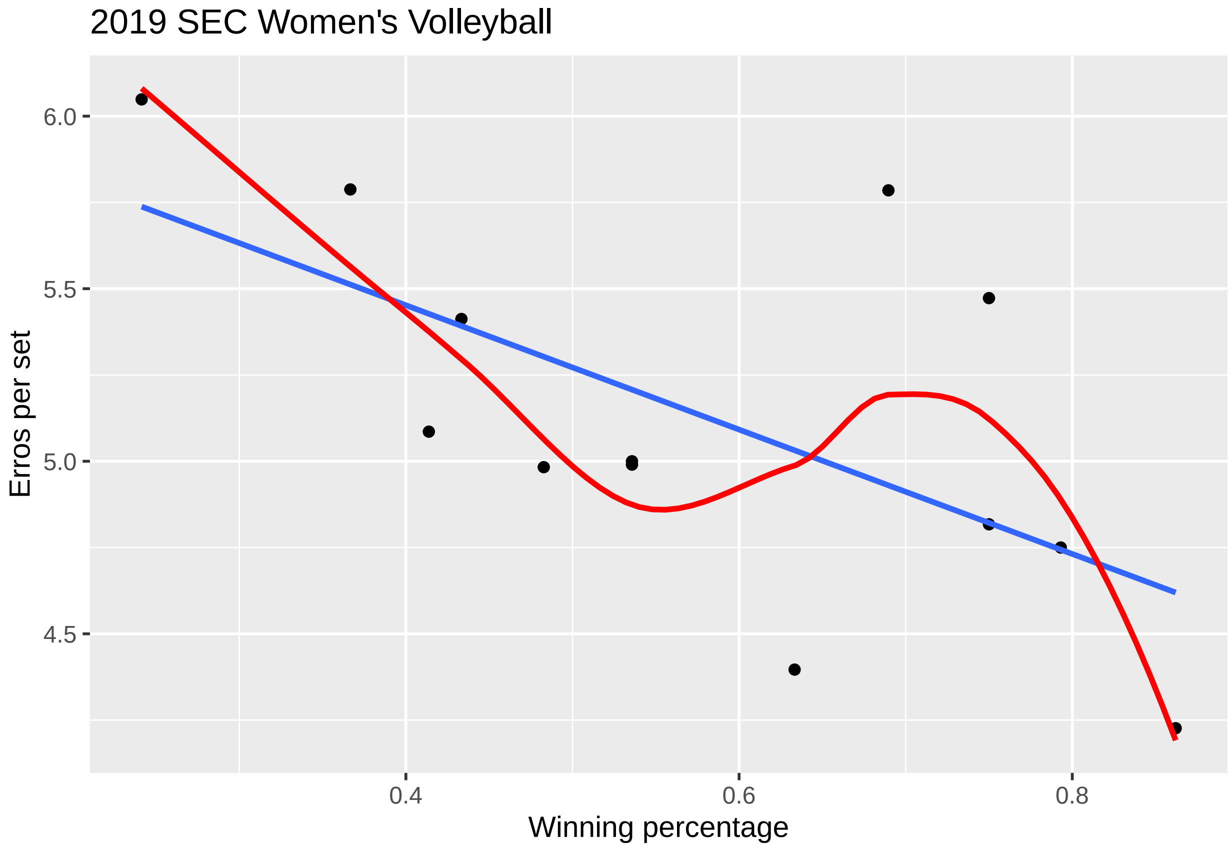
Answer: (a)

2

> Using the *volleyball-team-2019.csv* data to address this question. Create a plot that displays winning percentage vs. errors per set for the teams in the SEC conference. Include a straight line and a smooth line to the plot. Add descriptive labels to the x-axis and y-axis.

```r
SEC <- vb_team %>%
  filter(Conference == "SEC") %>%
  mutate(errors_per_set = Errors/Sets) %>%
  select(Team, Conference, Win_pct, errors_per_set, everything())

ggplot(SEC, aes(x = Win_pct, y = errors_per_set)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  geom_smooth(se = FALSE, color = "red") +
  xlab("Winning percentage") +
  ylab("Erros per set") +
  ggtitle("2019 SEC Women's Volleyball")
```



3

Using your SEC data from question 2, estimate the slope and intercept of a linear model fit to winning percentage as the response variable and errors per set as the explanatory variable. Compute the estimate slope and intercept using the regression formulas below and using the `lm()` function. How do the estimates using the two methods compare?

Slope:

$$\hat{a}_1 = r\frac{s_y}{s_x}$$

where r is the correlation between response variable y and explantory variable x, s_y is the standard deviation of y, and s_x is the standard deviation of x.

Intercept:

$$\hat{a}_0 = \bar{y} - \hat{a}_1\bar{x}$$

where $\bar{y}$ is the sample mean of y and $\bar{x}$ is the sample mean of x.

```r
#The estimate slope and intercept
x <- SEC$errors_per_set
y <- SEC$Win_pct
mx <- mean(x)
my <- mean(y)
sx <- sd(x)
sy <- sd(y)
r <- cor(x,y)
slope <- r *sy/sx
```

```
intercept <- my - slope*mx
slope
```

```
## [1] -0.2126626
```

```
intercept
```

```
## [1] 1.667985
```

```
#Using lm() function
get_estimates <- function(x,y)
{
  fit <- lm(y ~ x)
  return (fit)
}

get_estimates(x,y)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##      1.6680      -0.2127
```

The estimates using both approaches are very close to each others.

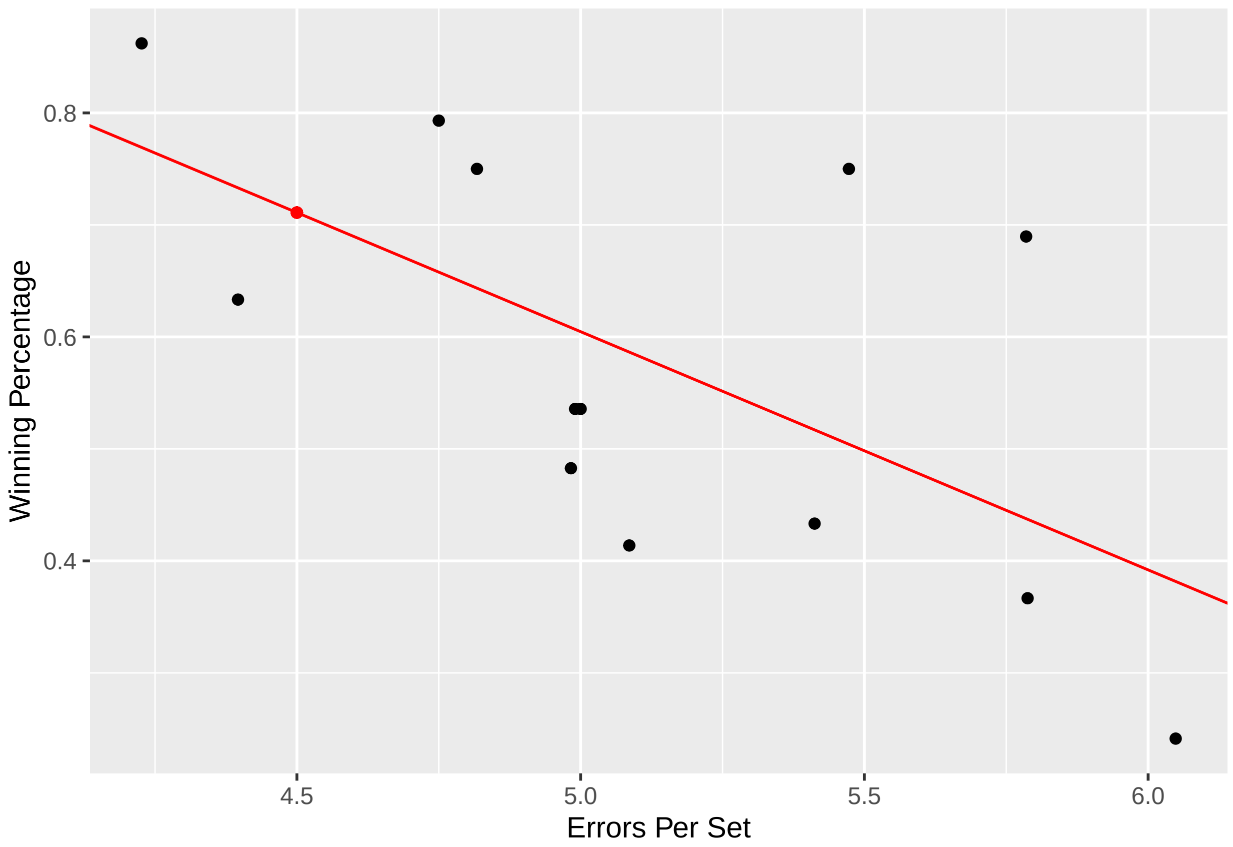
4

> Create a plot that displays winning percentage versus errors per set for the teams in the SEC conference. Add the regression model fit in the previous question (using the `lm()` method) to the plot. You may find `geom_abline()` useful for adding your fit model to the plot. Using this estimated model, predict the winning percentage for an SEC team that makes 4.5 errors per set. Plot this value as a red point on your plot.

```
fit_4<- lm(Win_pct~errors_per_set, data = SEC)
coef(fit_4)
```

```
##    (Intercept) errors_per_set
##      1.6679851     -0.2126626
```

```
ggplot(SEC, aes(errors_per_set, Win_pct)) +
  geom_point() +
  geom_abline(slope = -0.2126626 , intercept = 1.6679851, color ="red") +
  labs(x= "Errors Per Set", y = "Winning Percentage") +
  geom_point(x=4.5, y=(4.5*-0.2126626+1.667985), color="red")
```



5

> Using the *vb-division1-2019-all-matches.csv* data, find the match that occurred between Ole Miss and Alabama on 2019-11-03. Use the data from this match to answer this question.

We would like to fit a logistic model for the Ole Miss - Alabama match. Assume that this match is made up of a series of independent Bernoulli trials, each resulting in a point for one of the teams, until the match is won. Assume also that the chance of a team winning a point is the same whether or not the team is serving and assume that the chance does not depend on the current score of the game.

The Bernoulli probability mass function is

$$f(x|p) = p^x(1-p)^{1-x}$$

where x can equal 0 or 1 ($x = 1$ if Ole Miss gets the point and $x = 0$ if Alabama gets the point).

If we have n points scored among the two teams, $x_1, \ldots, x_n$, then we can write the likelihood as

$$L(p; x) = \Pi_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} = p^{\sum x_i}(1-p)^{n-\sum x_i}.$$

A simple model for the probability that Ole Miss with strength θ_1 wins a single point versus Alabama with strength θ_2 is a function of

$$\Delta = \theta_1 - \theta_2$$

where the probability of Ole Miss winning a point is

$$\mathsf{P}(\Delta) = \frac{1}{1 + \mathrm{e}^{-\Delta}}.$$

- The logistic function is a function of the log odds, $\ln(p/(1-p))$, where p is a probability.
  - That is, $\ln(p/(1-p)) = \Delta$.

  - If we solve for p we get the inverse-logistic function (displayed above for $P(\Delta)$).
- In summary, we model the Bernoulli probability p as a function of Δ (so we can write $p = P(\Delta)$). Ultimately we want the maximum likelihood estimate of p and of Δ.

Here are the steps to address this question.

- Write out the Bernoulli likelihood for the noted Ole Miss vs Alabama match

- Plot the likelihood function for p

- Compute the maximum likelihood estimate for p - you can do this mathematically or computationally

- Add a vertical red dashed line on your plot at the maximum likelihood value for p

- Compute the maximum likelihood value for Δ by putting the estimate for p into the log odds formula

```
vb_match %>%
  filter(team1=="Ole Miss" & team2 == "Alabama")
```

```
## # A tibble: 1 x 22
##   index date       team1   conference1 team2 conference2 site   s1_1  s1_2  s1_3
##   <int> <date>     <chr>   <chr>       <chr> <chr>       <chr> <dbl> <dbl> <dbl>
## 1  3906 2019-11-03 Ole Mi~ SEC         Alab~ SEC         <NA>     23    25    18
## # ... with 12 more variables: s1_4 <dbl>, s1_5 <dbl>, sets_1 <dbl>, s2_1 <dbl>,
## #   s2_2 <dbl>, s2_3 <dbl>, s2_4 <dbl>, s2_5 <dbl>, sets_2 <dbl>, winner <chr>,
## #   loser <chr>, attendance <dbl>
```

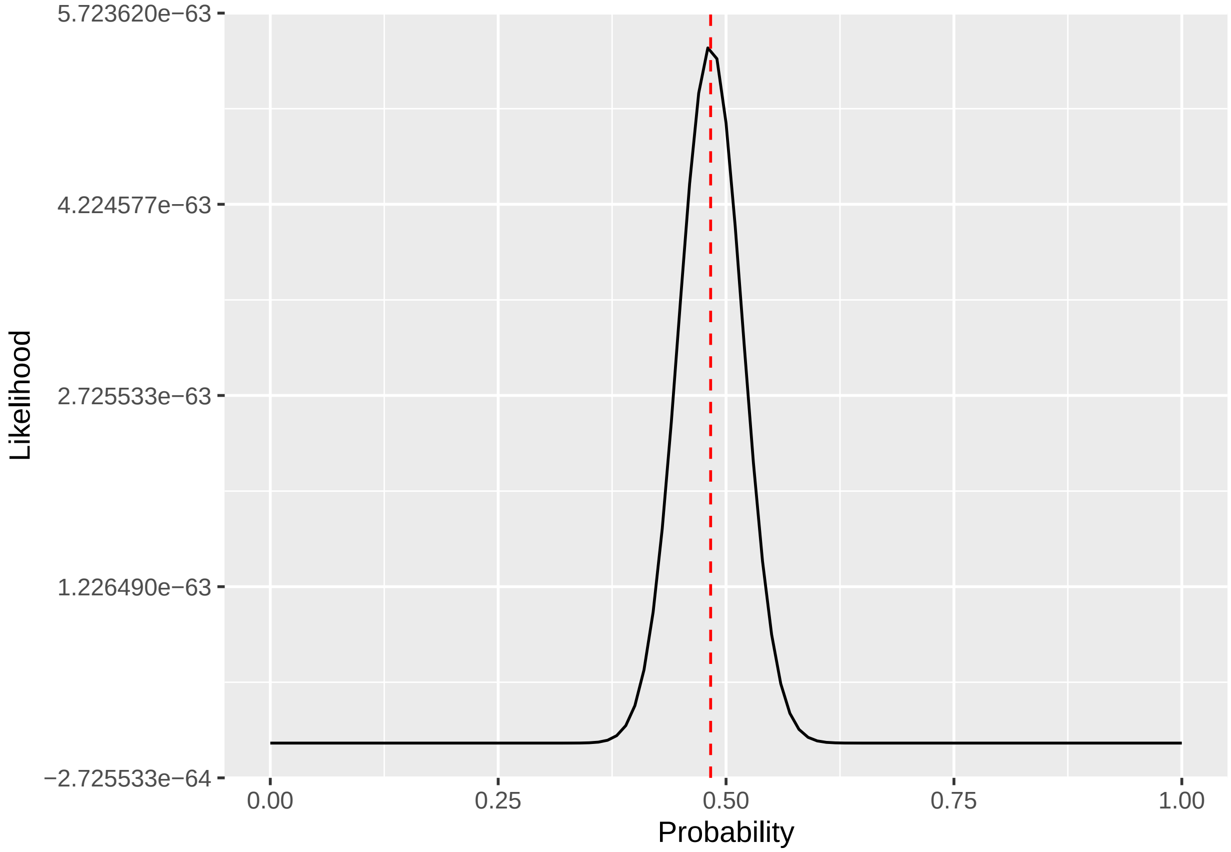
```
total_1 <- 23+25+18+25+9
total_2 <- 25+20+25+22+15
```

likelihood $= \mathsf{P}(\Delta)^{100}(1-\mathsf{P}(\Delta))^{107}$

```
Delta = seq(0,1 , by=0.01)

Bp= (Delta**100)*((1-Delta)**107)

ggplot(mapping = aes(x = Delta, y = Bp))+
  geom_line()+
  geom_vline(xintercept = 100/207, color="red", linetype = "dashed")+
  xlab("Probability") +
  ylab("Likelihood")
```



$\Delta = \ln(\frac{100}{207}/(1-\frac{100}{207}) = \ln(\frac{100}{207}/\frac{107}{207}) = -0.06765865$

```
p = 100/207
log(100/107)
```

```
## [1] -0.06765865
```