Homework 4

# 23.2.1 Exercises

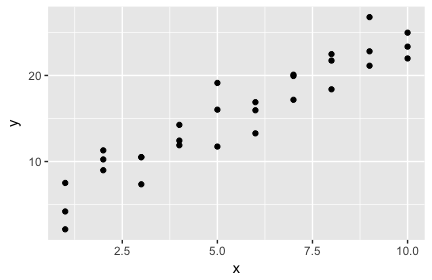
## (1) 23.2.1 Exercise 1

One downside of the linear model is that it is sensitive to unusual values because the distance incorporates a squared term. Fit a linear model to the simulated data below, and visualize the results. Rerun everything 6 times to generate different simulated datasets, and also examine the slope and intercept of the generated models, as well as the R2R2 values. How much variability do you see? How do the data values affect the model?

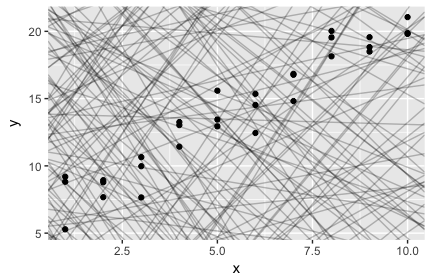
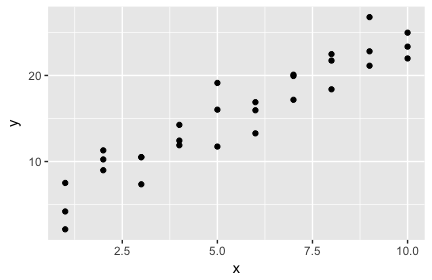
sim1a <- tibble(  
 x = rep(1:10, each = 3),  
 y = x \* 1.5 + 6 + rt(length(x), df = 2)  
)

Hint: To make it easy to do this 6 times, it makes sense to wrap everything into a function, and call that from within a for loop. The following code will help you accomplish this:

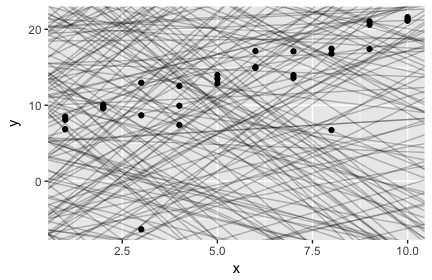
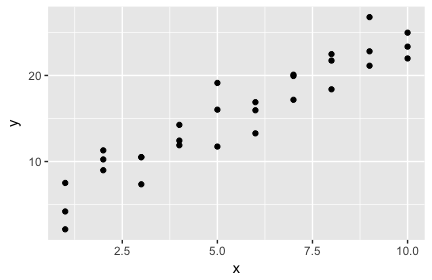
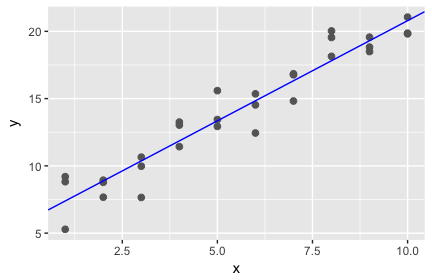
p <- ggplot(sim1, aes(x, y)) +   
 geom\_point()  
print(p)



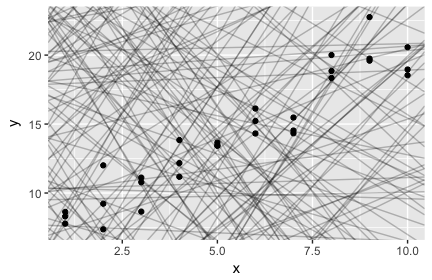
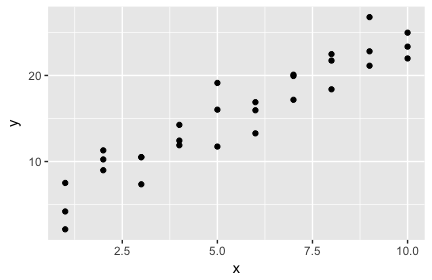
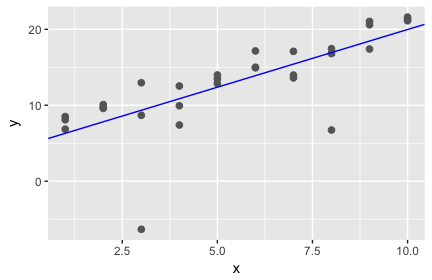
#Answer in a for loop  
for(i in 1:6){  
 sim1a <- tibble(  
 x=rep(1:10, each=3),  
 y=x\*1.5+6+rt(length(x), df=2)  
 )  
  
 p <- ggplot(sim1, aes(x, y)) +   
 geom\_point()  
 print(p)  
  
 models <- tibble(  
 a0 = runif(250, -20, 40),  
 a1 = runif(250, -5, 5)  
 )  
  
 p2 <- ggplot(sim1a, aes(x,y)) + geom\_abline(aes(intercept = a0, slope=a1), data = models, alpha = 1/4) + geom\_point()  
 print(p2)  
  
 sim1a\_mod <- lm(y ~x, data = sim1a)  
 coef(sim1a\_mod) %>% str()  
 p3 <- summary(sim1a\_mod)$r.squared \* 100  
 print(p3)  
   
 p4 <- ggplot(sim1a, aes(x, y)) +  
 geom\_point(size = 2, color = "grey40") +  
 geom\_abline(intercept = coef(sim1a\_mod)[1], slope = coef(sim1a\_mod)[2], color = "blue")  
 print(p4)  
   
 grid <- sim1a %>%  
 add\_predictions(sim1a\_mod)  
 grid  
}



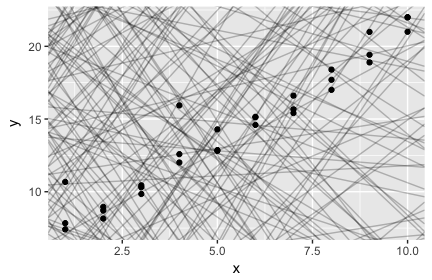
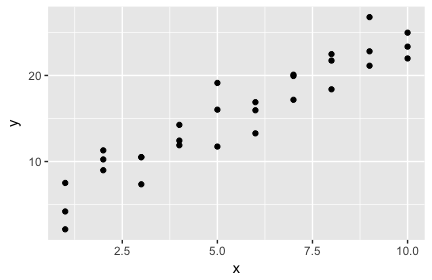
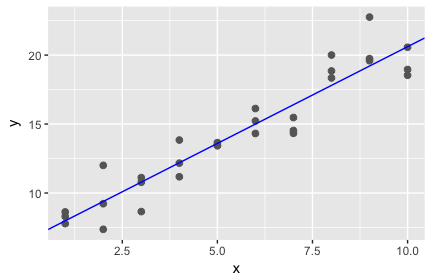
## Named num [1:2] 5.9 1.49  
## - attr(\*, "names")= chr [1:2] "(Intercept)" "x"  
## [1] 92.07506



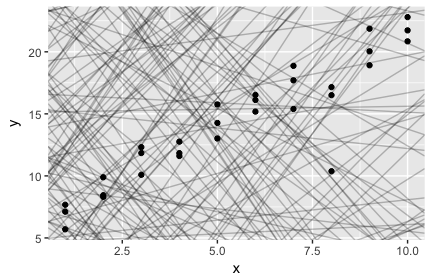
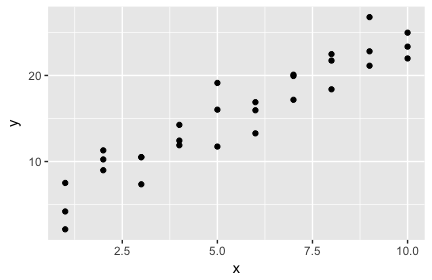
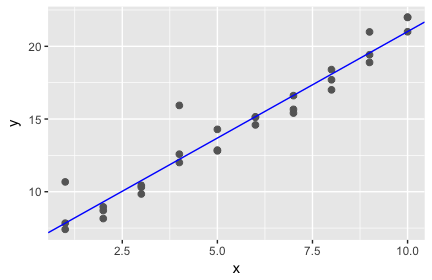
## Named num [1:2] 4.78 1.52  
## - attr(\*, "names")= chr [1:2] "(Intercept)" "x"  
## [1] 56.33763



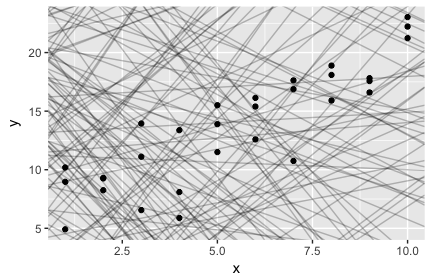
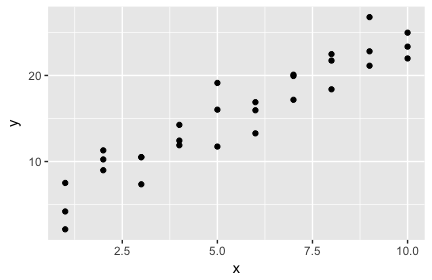
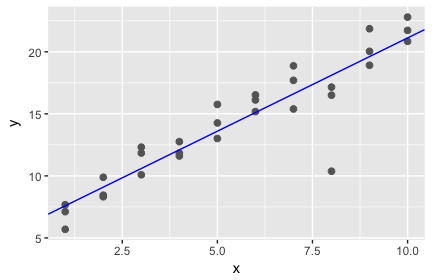
## Named num [1:2] 6.58 1.4  
## - attr(\*, "names")= chr [1:2] "(Intercept)" "x"  
## [1] 89.56837



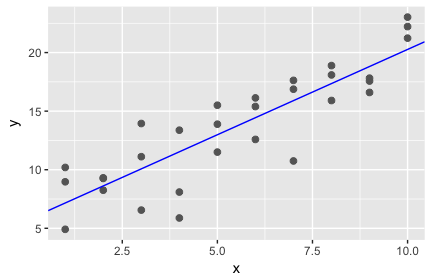
## Named num [1:2] 6.37 1.46  
## - attr(\*, "names")= chr [1:2] "(Intercept)" "x"  
## [1] 93.7321



## Named num [1:2] 6.09 1.5  
## - attr(\*, "names")= chr [1:2] "(Intercept)" "x"  
## [1] 84.88453



## Named num [1:2] 5.7 1.46  
## - attr(\*, "names")= chr [1:2] "(Intercept)" "x"  
## [1] 75.99046



There is some variability, but only to a certain extent. Slopes seem to be contained between the following (1< slope < 2), and the intercept varies between (4 < intercept < 7). This is for the observance of the data, but if we were to generate more datasets and especially if we were to change the method in which we are doing so, we would see different slopes and intercepts.

--Still need to write how the data affects the models

## (2) 23.2.1 Exercise 2

One way to make linear models more robust is to use a different distance measure. For example, instead of root-mean-squared (RMS) distance, you could use mean-absolute-deviation (MAD) distance:

measure\_distance <- function(a, data) {  
 diff <- data$y - make\_prediction(a, data)  
 mean(abs(diff))  
}

Use optim() to fit this model to the simulated data above and visually compare it to the linear model, again examining at least 6 instances. You want both models to be drawn on top of each instance of the simulated data set. Discuss what you find.

Hints: make\_prediction() is not a built-in function; you will need to create your own linear function. Unlike Exercise 1, here you should only compare the models graphically; I don’t think the R2R2 metric is meaningful for a model that uses the MAD distance measure.

# 23.3.3 Exercises

## (3) 23.3.3 Exercise 1

Instead of using lm() to fit a straight line, you can use loess() to fit a smooth curve. Repeat the process of model fitting, grid generation, predictions, and visualization on sim1 using loess() instead of lm(). How does the result compare to geom\_smooth()?

## (4) 23.3.3 Exercise 2

add\_predictions() is paired with gather\_predictions() and spread\_predictions(). Use both gather\_predictions() and spread\_predictions() to repeat the analysis with the two models used in problem (3) above. How do these three functions differ?

## (5) 23.3.3 Exercise 3

What does geom\_ref\_line() do? What package does it come from? Why is displaying a reference line in plots showing residuals useful and important?

## (6) 23.3.3 Exercise 4

Why might you want to look at a frequency polygon of absolute residuals? What are the pros and cons compared to looking at the raw residuals?

# 23.4.5 Exercises

## (7) 23.4.5 Exercise 1

What happens if you repeat the analysis of sim2 using a model without an intercept? What happens to the model equation? What happens to the predictions?

## (8) 23.4.5 Exercise 4

For sim4, which of mod1 and mod2 is better? The book author believes that mod2 does a slightly better job at removing patterns, but claims the effects are subtle. Examine the R2R2 values for the two models, and develop a plot that attempts to support this claim. In the end, how well does your plot support the claim? How likely is it that the claim is true?