hw4

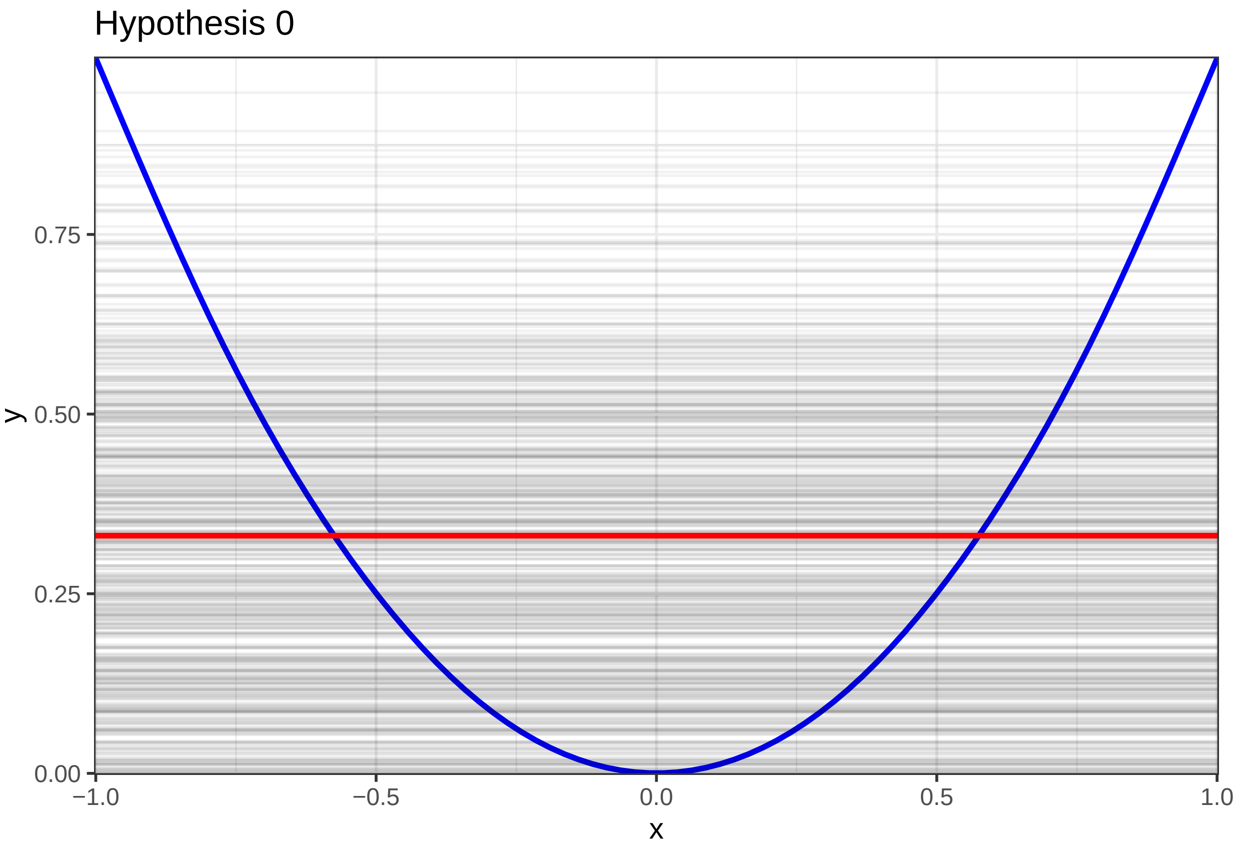
Kaicheng Luo

2019/9/30

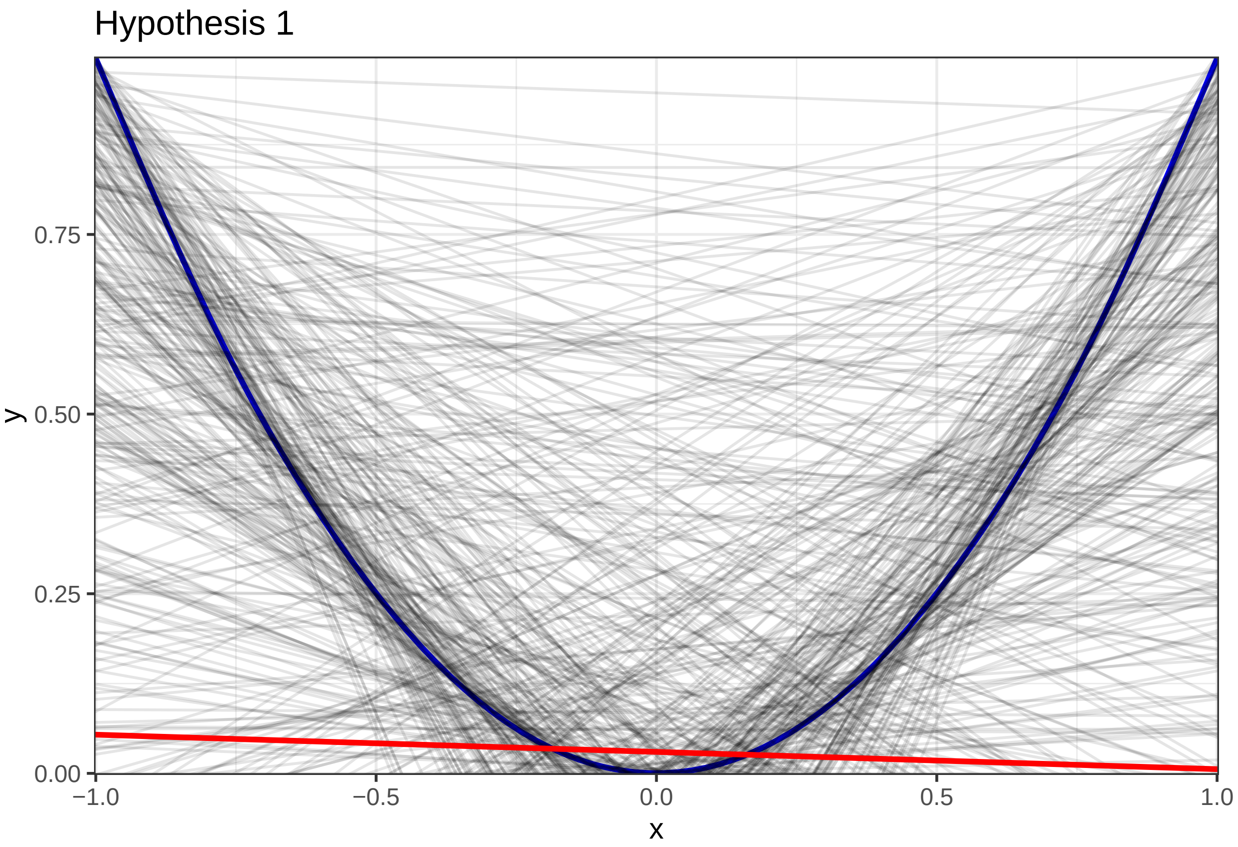
Simulation

```
# Model H0: y = coef_0
# Model H1: y = coef_1a + coef_1b
n_out <- 20000
epsilon <- 0

# For each iteration, I'd like to record the following values
coef_0 <- vector()
coef_1a <- vector()
coef_1b <- vector()
MSE_0 <- rep(0, 500)
MSE_1 <- rep(0, 500)

# Here we randomly draw a test set
set.seed(12345)
x_out <- runif(n_out, min = -1, max = 1)
y_out <- x_out^2 + epsilon

# Fit the model with 2 training sets, and calculate the MSE of each model we fit
for (i in 1:500){
  x <- runif(n = 2, min = -1, max = 1)
  y <- x^2 + epsilon
  coef_0 <- c(coef_0, (y[1]+y[2])/2)
  coef_1a <- c(coef_1a, y[1] - x[1]*(y[2]-y[1])/(x[2]-x[1]))
  coef_1b <- c(coef_1b, (y[2]-y[1])/(x[2]-x[1]))
  for (j in 1:n_out){
    MSE_0[i] = MSE_0[i] + ((y_out[j] - coef_0[i])^2)/n_out
    MSE_1[i] = MSE_1[i] + ((y_out[j] - coef_1a[i] - coef_1b[i] * x_out[j])^2)/n_out
  }
}
```

```
# Here's the code for the H_0 plot
plot_data <- data.frame(x = x_out, y = y_out)
plot_data %>%
  ggplot() + theme_bw() +
  geom_smooth(aes(x = x, y = y), color = 'blue') +
  geom_hline(yintercept = coef_0, alpha = 0.05) +
  geom_hline(yintercept = mean(coef_0), color = 'red', size = 1) +
  scale_y_continuous(expand = c(0.001, 0.001)) +
  scale_x_continuous(expand = c(0.001, 0.001)) +
  labs(
    title = "Hypothesis 0"
  )
```



```
# Here's the code for the H_1 plot
plot_data %>%
  ggplot() + theme_bw() +
  geom_smooth(aes(x = x, y = y), color = 'blue') +
  geom_abline(intercept = coef_1a, slope = coef_1b, alpha = 0.1) +
  geom_abline(intercept = mean(coef_1a), slope = mean(coef_1b), color = 'red', size = 1) +
  scale_y_continuous(expand = c(0.001, 0.001)) +
  scale_x_continuous(expand = c(0.001, 0.001)) +
  labs(
    title = "Hypothesis 1"
  )
```



```
# To verify the bias-variance trade-off rigorously, we examine the first model
OverallMSE_0 <- mean(MSE_0)
bias <- vector()
variance <- vector()
for (i in 1:20000){
  bias[i] <- (mean(coef_0) - y_out[i])^2
}
bias_0 <- mean(bias)
for (i in 1:500){
  variance[i] <- (mean(coef_0) - coef_0[i])^2
}
variance_0 <- mean(variance)
# Display it tidily
model_0 <- data.frame("OverallMSE" = OverallMSE_0, "Bias" = bias_0, "Var" = variance_0)
model_0
```

```
##   OverallMSE       Bias        Var
## 1  0.1330587 0.08903894 0.04401978
```

```
# If the decomposition is correct, it shall return 0
model_0$OverallMSE - model_0$Bias - model_0$Var
```

```
## [1] 6.938894e-18
```

```r
# Similarly, we can do this to the second model
OverallMSE_1 <- mean(MSE_1)
bias <- vector()
variance <- rep(0, n_out)
for (i in 1:20000){
  bias[i] <- (mean(coef_1a) + mean(coef_1b)*x_out[i] - y_out[i])^2
}
bias_1 <- mean(bias)
for (i in 1:500){
  for (j in 1:n_out){
    variance[i] = variance[i] + ((mean(coef_1a) + mean(coef_1b)*x_out[j] - coef_1a[i] - coef_1b[i]*x_ou
  }
}

variance_1 <- mean(variance)
# Display it tidily
model_1 <- data.frame("OverallMSE" = OverallMSE_1, "Bias" = bias_1, "Var" = variance_1)
model_1
```

```
##   OverallMSE      Bias       Var
## 1   0.486243 0.1802108 0.3060323
```

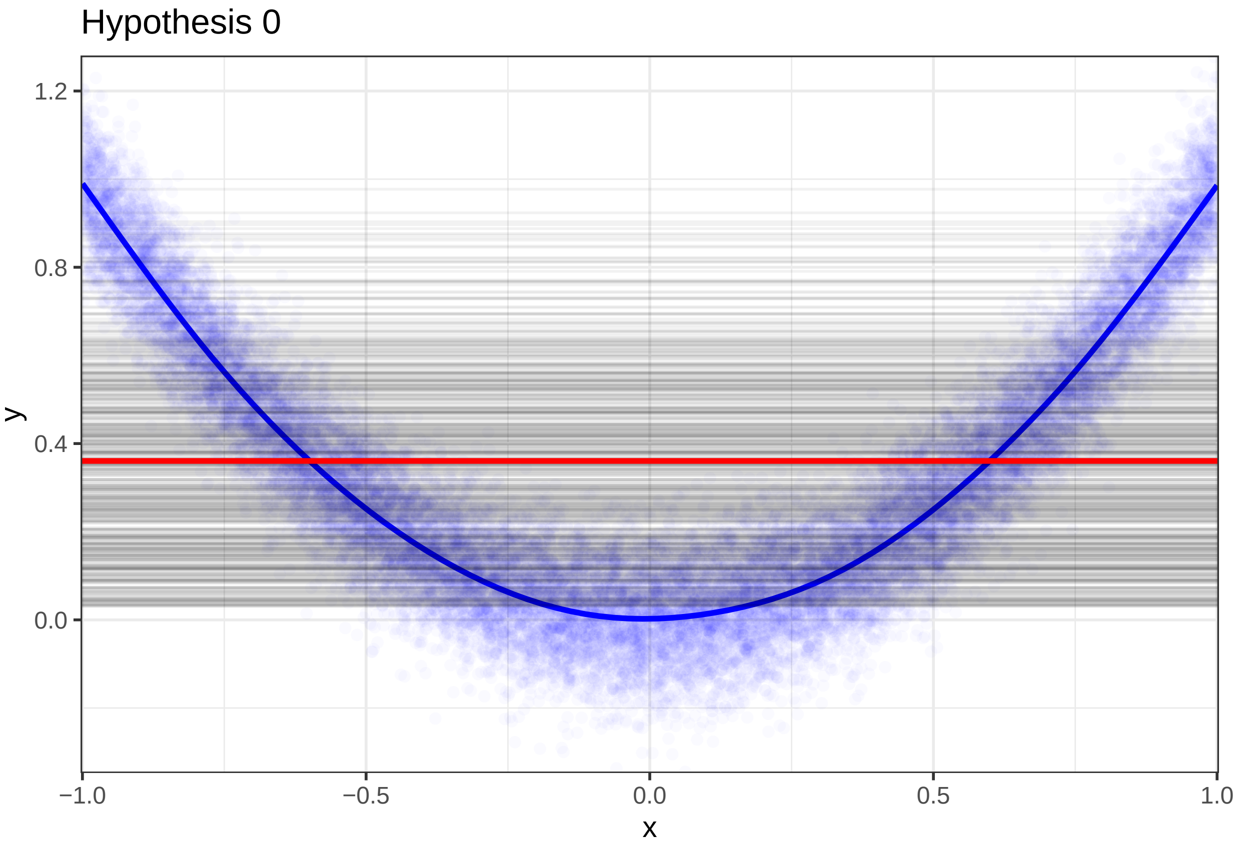
```r
# If the decomposition is correct, it shall return 0
model_1$OverallMSE - model_1$Bias - model_1$Var
```

```
## [1] 0
```

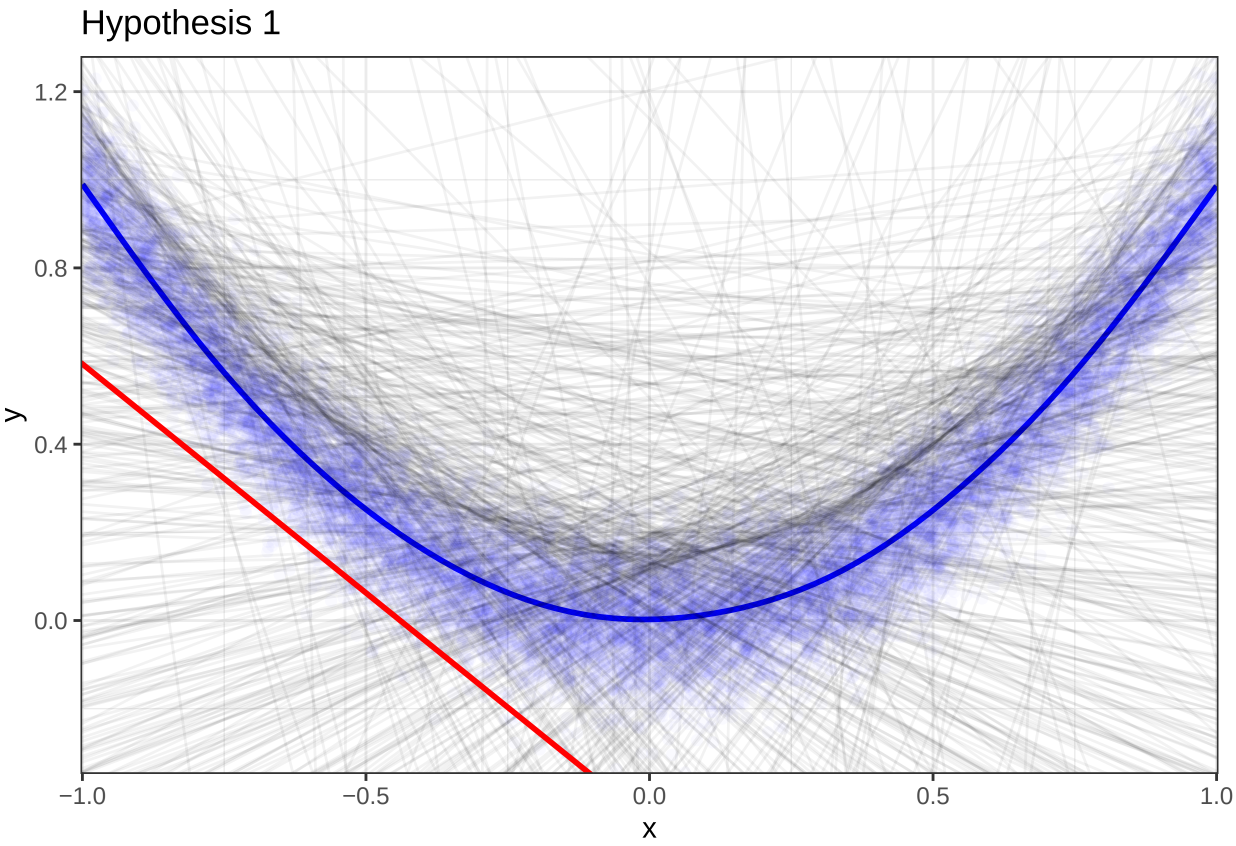
Run another simulation with noise

```r
# Model H0: y = coef_0
# Model H1: y = coef_1a + coef_1b
n_out <- 20000
epsilon <- rnorm(n_out, 0, 0.1)

# For each iteration, I'd like to record the following values
coef_0 <- vector()
coef_1a <- vector()
coef_1b <- vector()
MSE_0 <- rep(0, 500)
MSE_1 <- rep(0, 500)

# Here we randomly draw a test set
set.seed(12345)
x_out <- runif(n_out, min = -1, max = 1)
y_out <- x_out^2 + epsilon

# Fit the model with 2 training sets, and calculate the MSE of each model we fit
for (i in 1:500){
  x <- runif(n = 2, min = -1, max = 1)
  y <- x^2 + epsilon
  coef_0 <- c(coef_0, (y[1]+y[2])/2)
  coef_1a <- c(coef_1a, y[1] - x[1]*(y[2]-y[1])/(x[2]-x[1]))
  coef_1b <- c(coef_1b, (y[2]-y[1])/(x[2]-x[1]))
  for (j in 1:n_out){
    MSE_0[i] = MSE_0[i] + ((y_out[j] - coef_0[i])^2)/n_out
    MSE_1[i] = MSE_1[i] + ((y_out[j] - coef_1a[i] - coef_1b[i] * x_out[j])^2)/n_out
  }
}
```

```
# Here's the code for the H_0 plot
plot_data <- data.frame(x = x_out, y = y_out)
plot_data %>%
  ggplot() + theme_bw() +
  geom_point(aes(x = x, y = y), color = 'blue', alpha = 0.02) +
  geom_smooth(aes(x = x, y = y), color = 'blue') +
  geom_hline(yintercept = coef_0, alpha = 0.05) +
  geom_hline(yintercept = mean(coef_0), color = 'red', size = 1) +
  scale_y_continuous(expand = c(0.001, 0.001)) +
  scale_x_continuous(expand = c(0.001, 0.001)) +
  labs(
    title = "Hypothesis 0"
  )
```



```
# Here's the code for the H_1 plot
plot_data %>%
  ggplot() + theme_bw() +
  geom_smooth(aes(x = x, y = y), color = 'blue') +
  geom_point(aes(x = x, y = y), color = 'blue', alpha = 0.02) +
  geom_abline(intercept = coef_1a, slope = coef_1b, alpha = 0.05) +
  geom_abline(intercept = mean(coef_1a), slope = mean(coef_1b), color = 'red', size = 1) +
  scale_y_continuous(expand = c(0.001, 0.001)) +
  scale_x_continuous(expand = c(0.001, 0.001)) +
  labs(
    title = "Hypothesis 1"
  )
```



```
# To verify the bias-variance trade-off rigorously, we examine the first model
OverallMSE_0 <- mean(MSE_0)
bias <- vector()
variance <- vector()
for (i in 1:20000){
  bias[i] <- (mean(coef_0) - x_out[i]^2)^2
}
bias_0 <- mean(bias)
for (i in 1:500){
  variance[i] <- (mean(coef_0) - coef_0[i])^2
}
variance_0 <- mean(variance)
# Display it tidily
model_0 <- data.frame("OverallMSE" = OverallMSE_0, "Bias" = bias_0, "Var" = variance_0)
model_0
```

```
##   OverallMSE       Bias        Var
## 1  0.1430415 0.08987592 0.04401978
```

```r
# If the decomposition is correct, it shall return 0
model_0$OverallMSE - model_0$Bias - model_0$Var
```

```
## [1] 0.009145751
```

```r
OverallMSE_1 <- mean(MSE_1)
bias <- vector()
variance <- rep(0, n_out)
for (i in 1:20000){
  bias[i] <- (mean(coef_1a) + mean(coef_1b)*x_out[i] - x_out[i]^2)^2
}
bias_1 <- mean(bias)
for (i in 1:500){
  for (j in 1:n_out){
    variance[i] = variance[i] + ((mean(coef_1a) + mean(coef_1b)*x_out[j] - coef_1a[i] - coef_1b[i]*x_ou
  }
}

variance_1 <- mean(variance)
# Display it tidily
model_1 <- data.frame("OverallMSE" = OverallMSE_1, "Bias" = bias_1, "Var" = variance_1)
model_1
```

```
##   OverallMSE     Bias      Var
## 1    298.159 1.061858 297.0879
```

```r
# If the decomposition is correct, it shall return 0
model_1$OverallMSE - model_1$Bias - model_1$Var - 0.01
```

```
## [1] -0.000808448
```