HW6 Key

Andy Hoegh

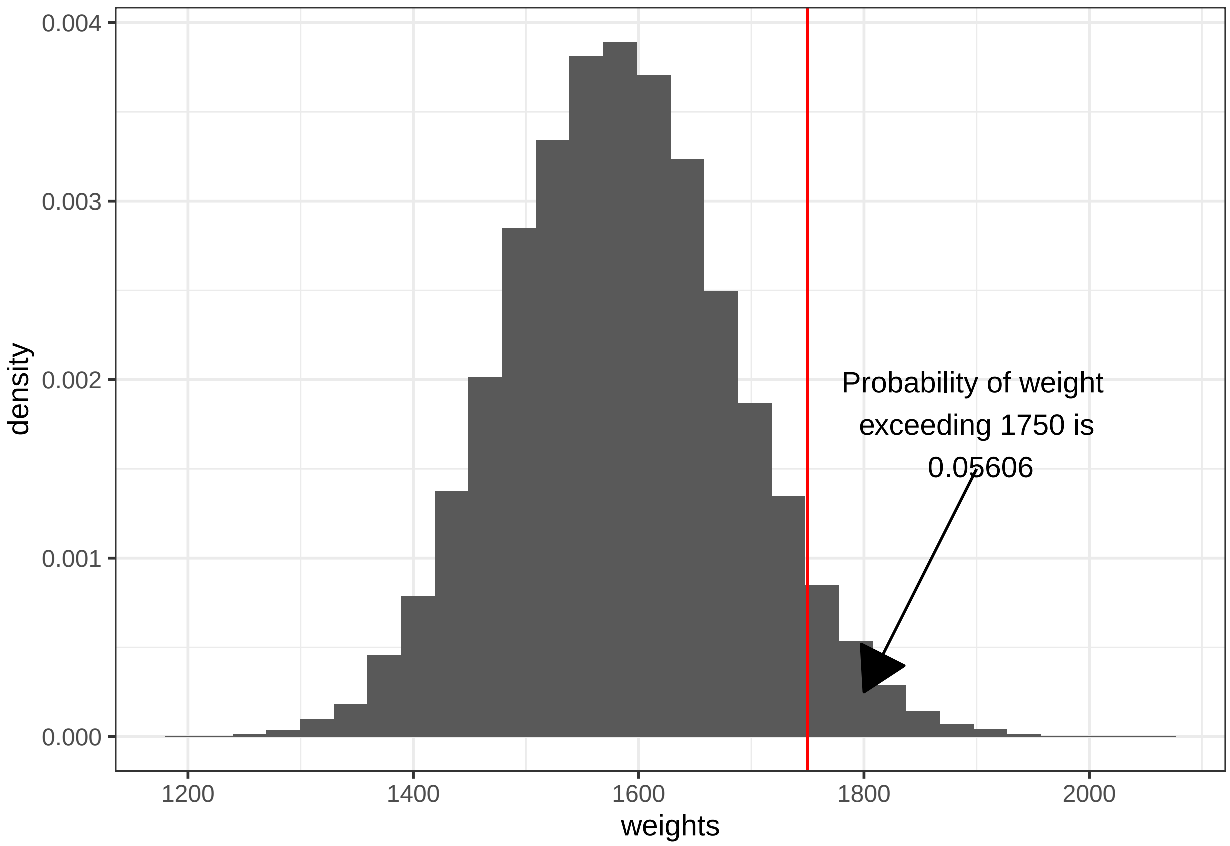
HW6

1. 4 points (Based on ROS 5.2)

The logarithms of weights (in pounds) of men in the United States are approximately normally distributed with mean 5.13 and standard deviation of 0.17; women's log weights are approximately normally distributed with mean 4.96 and standard deviation of 0.20. Suppose 10 adults selected at random step on an elevator with a capacity of 1750 pounds. What is the probability that their total weight exceeds this limit?

```
mu <- c(5.13, 4.96)
sd <- c(0.17, 0.20)

num_reps <- 50000
weights <- rep(0, num_reps)

for (i in 1:num_reps){
  who <- sample(2, 10, replace = T)
  weights[i] <- sum(exp(rnorm(10, mu[who], sd[who] )))
}

tibble(weights = weights) %>% ggplot(aes(x = weights)) +
  geom_histogram((aes(y = ..density..))) +
  theme_bw() + geom_vline(xintercept = 1750, color = 'red') +
  annotate("segment", x = 1900, y = .0015, xend = 1800, yend = .00025,
         arrow = arrow(type = "closed")) +
  annotate('text',x = 1900, y = .00175, label = paste('Probability of weight \nexceeding 1750 is\n', mea
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Clearly state any assumptions that you make in calculating this probability.

Assuming 50/50 for male / female and that the weight of each individual is independent of other individuals.

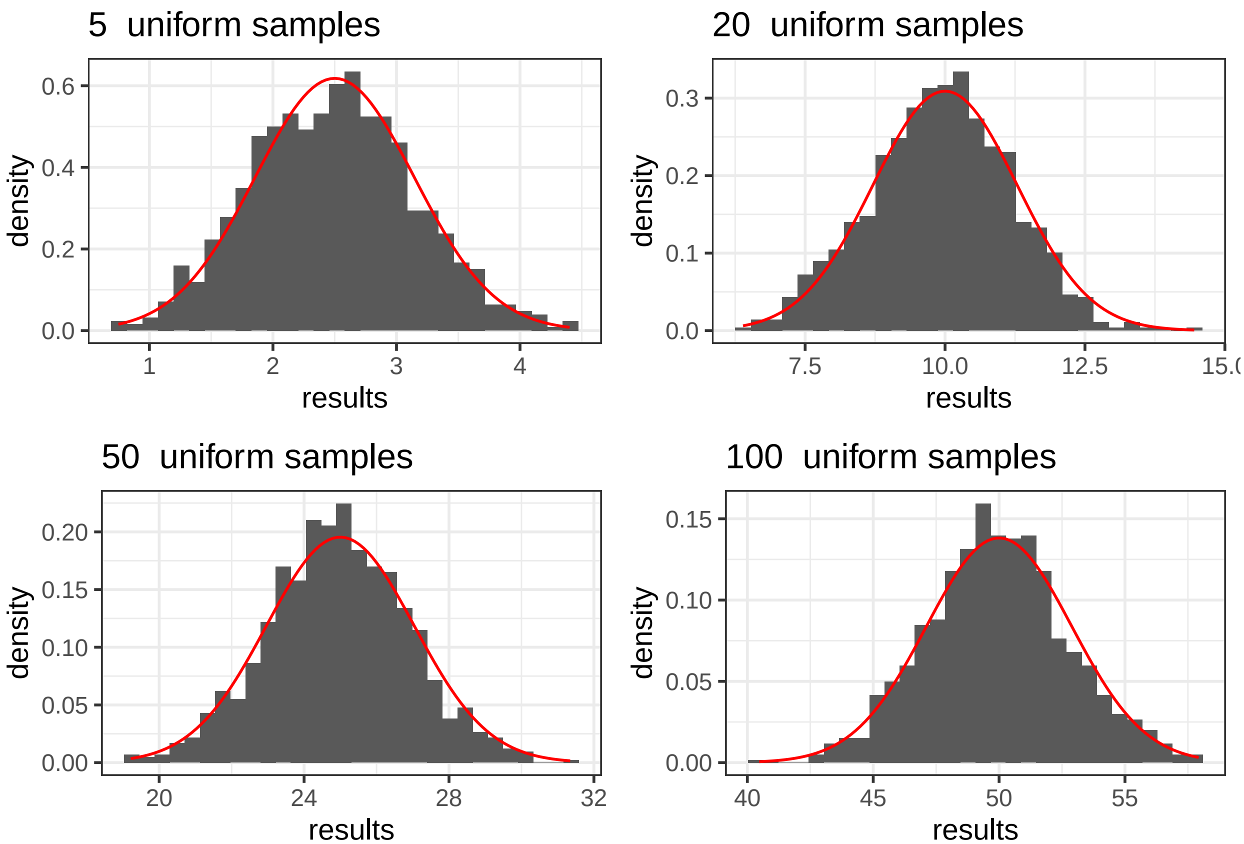
2. 4 points (Based on ROS 5.4)

For the following values of n = (5, 20, 50, 100), let x = x1 + … + xn, the sum of n independent uniform random variables. In R, create 1000 simulations of x (for each n) and plot their histogram. For each n, what is the normal approximation from the CLT (note that the variance of a uniform random variable is $\frac{1}{12}$ (b-a)^2\$, where b and a are the upper and lower bounds of the uniform variable). Overlay the normal density on top of each histogram and comment on any differences between the histogram and curve.

```
num_reps <- 1000
n <- 5
results <- rowSums(matrix(runif(num_reps * n), num_reps, n))

f5 <- tibble(results = results, each = num_reps) %>% ggplot(aes(x=results)) +
 geom_histogram(aes (y = ..density..)) + theme_bw() +
stat_function(fun = dnorm, args = list(mean = n/2,
                            sd = sqrt(n/12)), col = "red") +
  ggtitle(paste(n, ' uniform samples'))

n <- 20
results <- rowSums(matrix(runif(num_reps * n), num_reps, n))

f20 <- tibble(results = results, each = num_reps) %>% ggplot(aes(x=results)) +
 geom_histogram(aes (y = ..density..)) + theme_bw() +
```

```r
stat_function(fun = dnorm, args = list(mean = n/2,
                            sd = sqrt(n/12)), col = "red")  +
  ggtitle(paste(n, ' uniform samples'))

n <- 50
results <- rowSums(matrix(runif(num_reps * n), num_reps, n))

f50 <- tibble(results = results, each = num_reps) %>% ggplot(aes(x=results)) +
 geom_histogram(aes (y = ..density..)) + theme_bw() +
stat_function(fun = dnorm, args = list(mean = n/2,
                            sd = sqrt(n/12)), col = "red") +
  ggtitle(paste(n, ' uniform samples'))

n <- 100
results <- rowSums(matrix(runif(num_reps * n), num_reps, n))

f100 <- tibble(results = results, each = num_reps) %>% ggplot(aes(x=results)) +
 geom_histogram(aes (y = ..density..)) + theme_bw() +
stat_function(fun = dnorm, args = list(mean = n/2,
                            sd = sqrt(n/12)), col = "red") +
  ggtitle(paste(n, ' uniform samples'))

grid.arrange(f5, f20, f50, f100)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

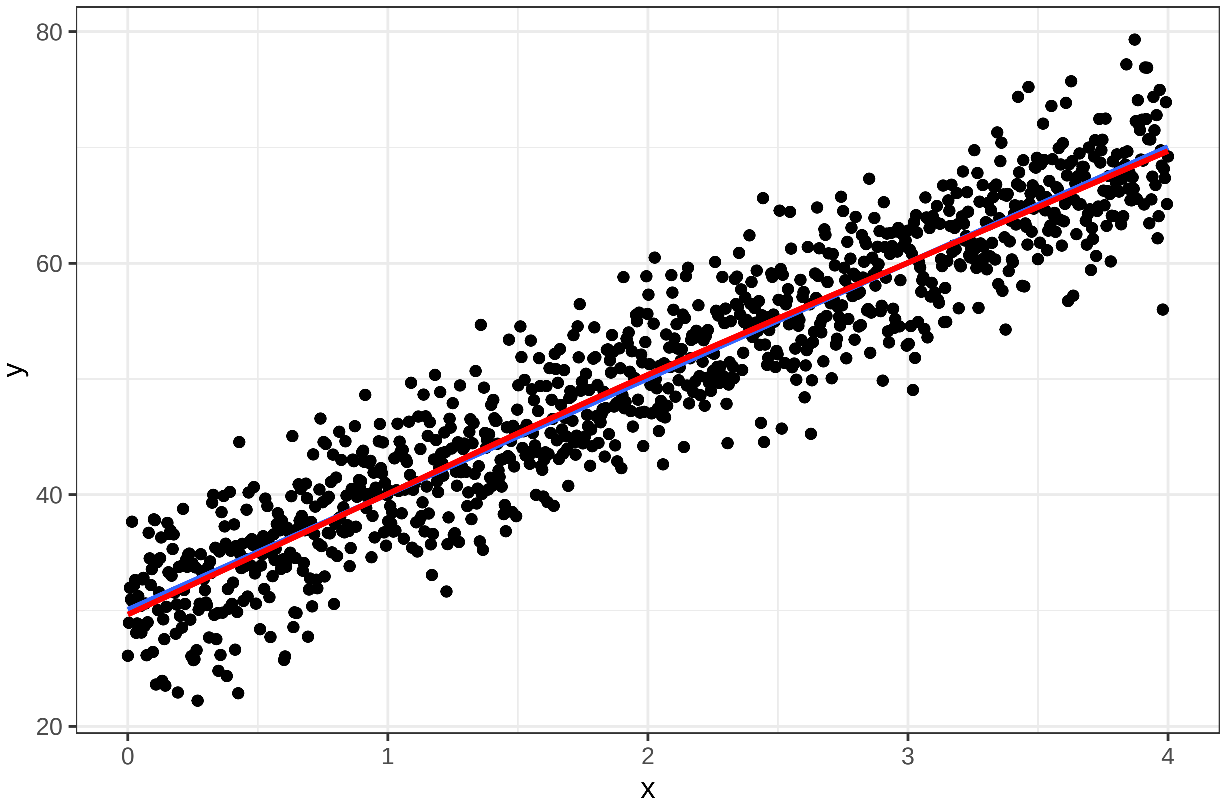


3. 3 points

Simulate and plot synthetic data with:

- x in the range of 0 to 4 percent corresponding to the regression line with y = 30 + 10 x, with residual standard deviation of 3.9

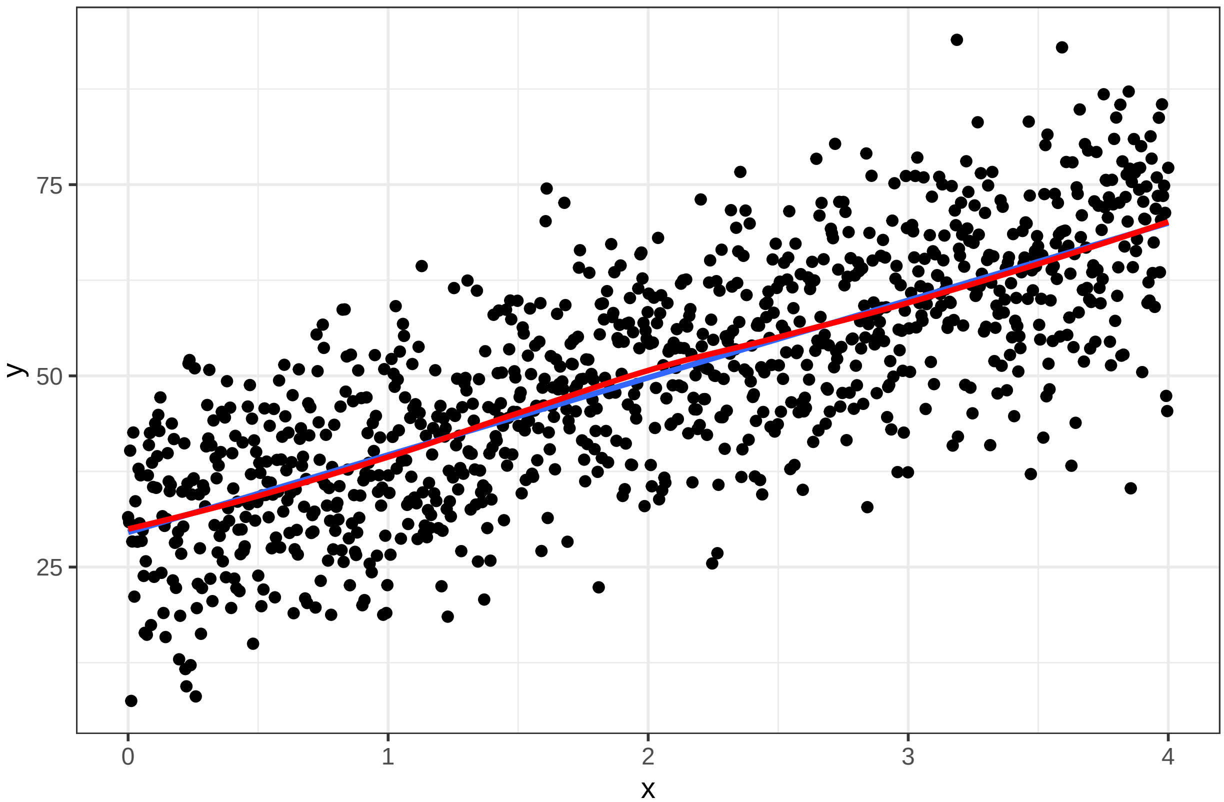
```
n <- 1000
x<- seq(0,4, length.out = n)
y1 <- rnorm(n, 30 + 10 * x, 3.9)
d1 <- tibble(x = x, y = y1)
d1 %>% ggplot(aes(y = y, x = x)) + geom_point() +
  theme_bw() +
  geom_smooth(formula = y ~x, method = 'lm', se = F) +
  geom_smooth(formula = y ~x, method = 'loess', color = 'red', se = F) +
  labs(caption= 'Loess fit in red, lm fit in blue')
```



Loess fit in red, lm fit in blue

- x in the range of 0 to 4 percent corresponding to the regression line with y = 30 + 10 x, with residual standard deviation of 10

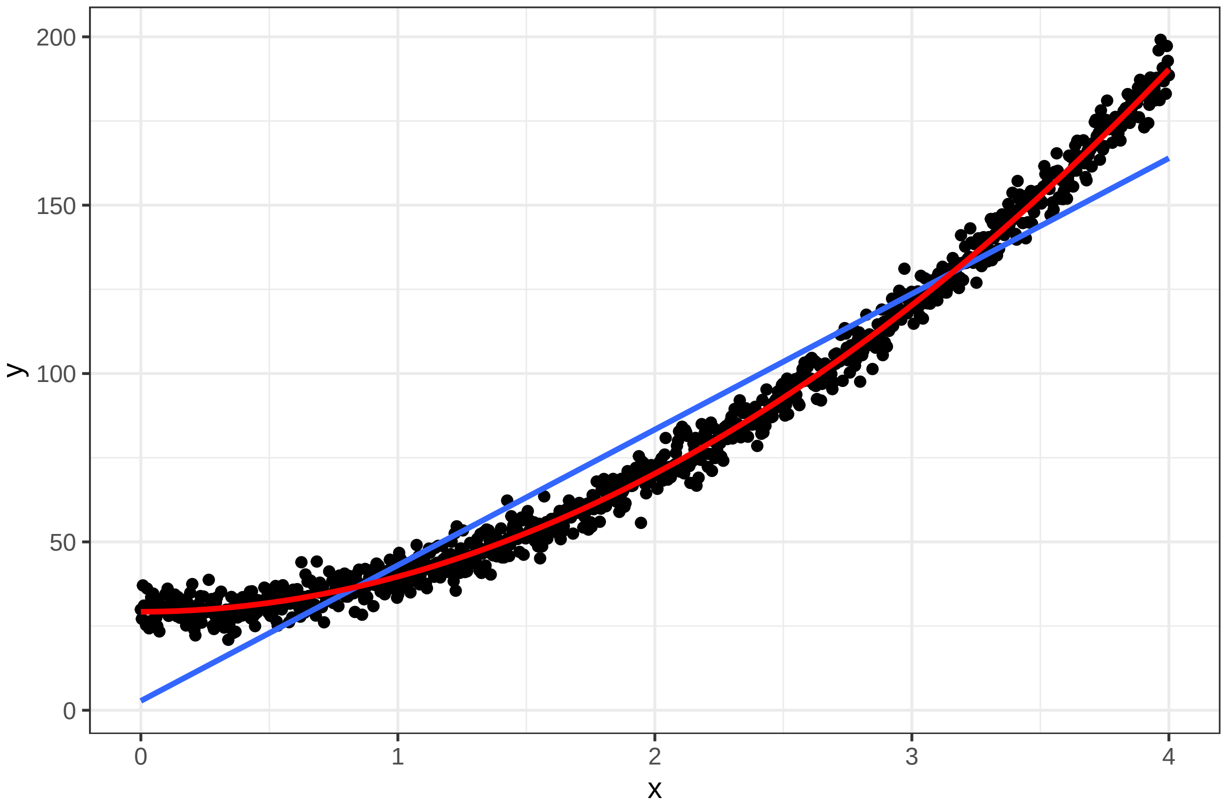
```
y2 <- rnorm(n, 30 + 10 * x, 10)
d2 <- tibble(x = x, y = y2)
d2 %>% ggplot(aes(y = y, x = x)) + geom_point() +
  theme_bw() +
  geom_smooth(formula = y ~x, method = 'lm', se = F) +
  geom_smooth(formula = y ~x, method = 'loess', color = 'red', se = F) +
  labs(caption= 'Loess fit in red, lm fit in blue')
```



Loess fit in red, lm fit in blue

x in the range of 0 to 4 percent corresponding to the regression line with y = 30 + 10 x^2, with residual standard deviation of 3.9

```
y3 <- rnorm(n, 30 + 10 * x^2, 3.9)
d3 <- tibble(x = x, y = y3, x_sq = x^2)
d3 %>% ggplot(aes(y = y, x = x)) + geom_point() +
  theme_bw() +
  geom_smooth(formula = y ~x, method = 'lm', se = F) +
  geom_smooth(formula = y ~x, method = 'loess', color = 'red', se = F) +
  labs(caption= 'Loess fit in red, lm fit in blue')
```



Loess fit in red, lm fit in blue

For each plot include the best linear fit, geom_smooth(method = 'lm'), as well as the LOESS fit, geom_smooth(method = 'loess')

4. 4 points

For each of the scenarios in Question 3, fit a linear regression model using either `lm` or `stan_glm`. For the third scenario fit one model with `y~x` and `y~x_squared`. For each situation, summarize the model fit and discuss how the results compare with your expectations.

```
stan_glm(y~x, data = d1, refresh = 0) %>% print()
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x
##  observations: 1000
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 30.1    0.3
## x           10.0    0.1
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 4.0    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
```

```
## * For info on the priors used see ?prior_summary.stanreg
```

Estimates look close to true values, standard errors are relatively small.

```r
stan_glm(y~x, data = d2, refresh = 0) %>% print()
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x
##  observations: 1000
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 29.6    0.6
## x           10.1    0.3
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 9.9    0.2
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Estimates look close to true values, standard errors are larger than previous dataset.

```r
stan_glm(y~x, data = d3, refresh = 0) %>% print()
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x
##  observations: 1000
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept)  2.8    0.8
## x           40.3    0.3
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 12.4    0.3
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```r
stan_glm(y~x_sq, data = d3, refresh = 0) %>% print()
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x_sq
##  observations: 1000
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 29.7    0.2
## x_sq        10.1    0.0
```

```
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 3.8    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Poor fit with linear relationship. Squared relationship is inline with true values.