Assignment 4a

Satvik Saha

2024-09-24

Answer 1

(a)

```
n <- 100
a <- 1      # intercept
b <- 2      # slope
sigma <- 3  # residual standard deviation
x <- runif(n, min = 0, max = 4)
y <- a + b*x + sigma * rnorm(n)
df <- data.frame(x, y)

library(rstanarm)
fit.y <- stan_glm(y ~ x, data = df, refresh = 0)
print(fit.y)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y ~ x
##  observations: 100
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 1.1    0.6
## x           2.1    0.3
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 3.1    0.2
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

To check whether or not the true coefficients lie within one standard deviation of the estimates, we do the following.

```
a.hat <- coef(fit.y)[1]
b.hat <- coef(fit.y)[2]
a.se <- se(fit.y)[1]
b.se <- se(fit.y)[2]

abs(a.hat - a) < a.se
```

```
## (Intercept)
```

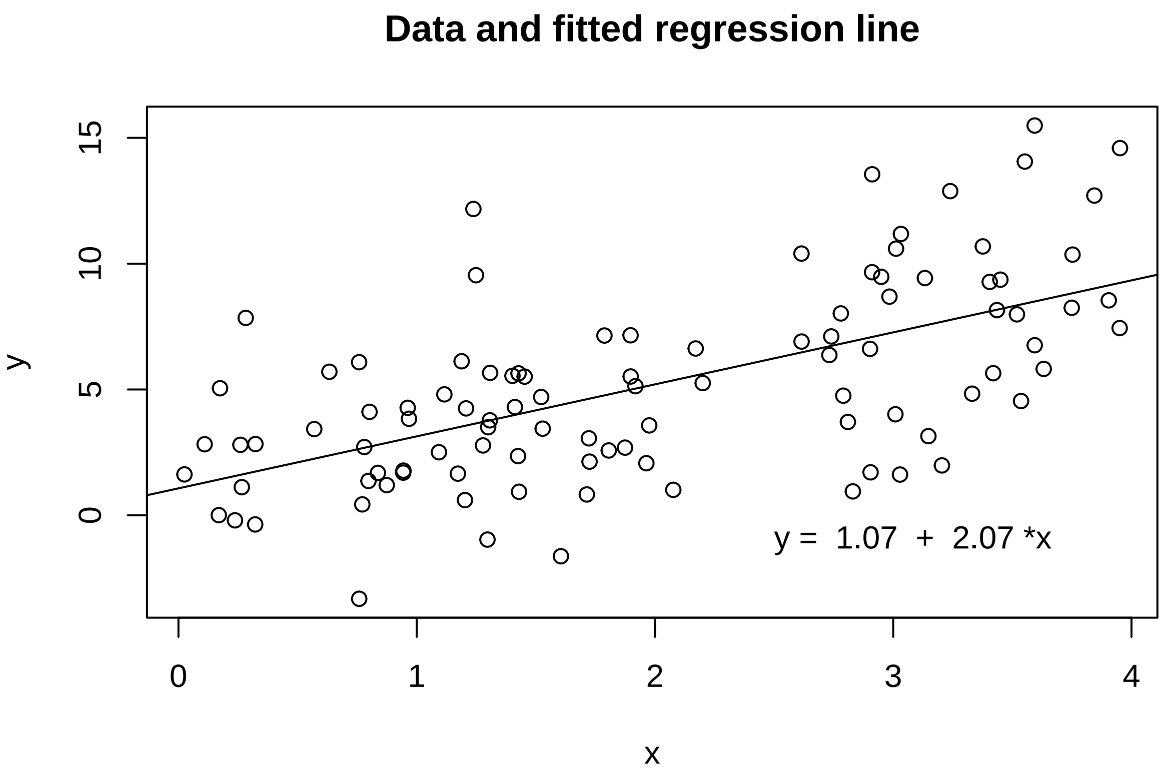
```
##        TRUE
```

```r
abs(b.hat - b) < b.se
```

```
##    x
## TRUE
```

(b)

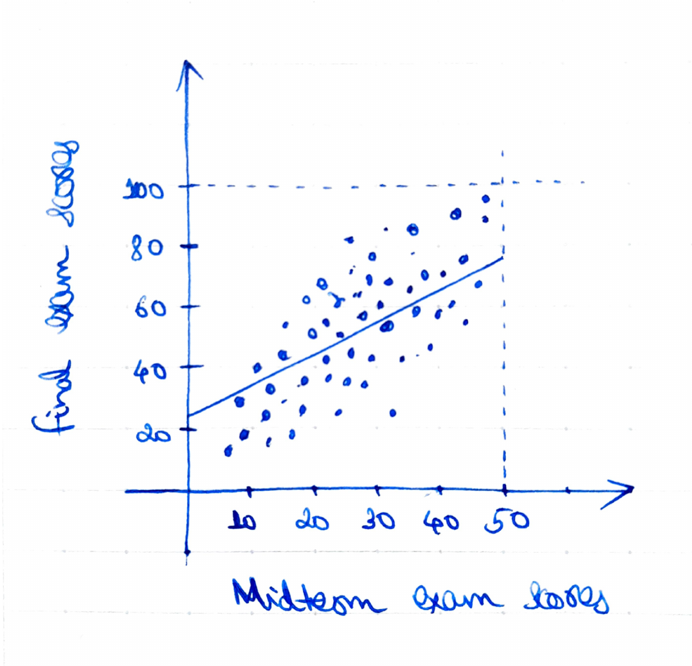
```r
plot(x, y, main = "Data and fitted regression line")
abline(a.hat, b.hat)
x.bar <- mean(x)
text(2.5, -1, paste("y = ", round(a.hat, 2), " + ", round(b.hat, 2), "*x"), adj = 0)
```

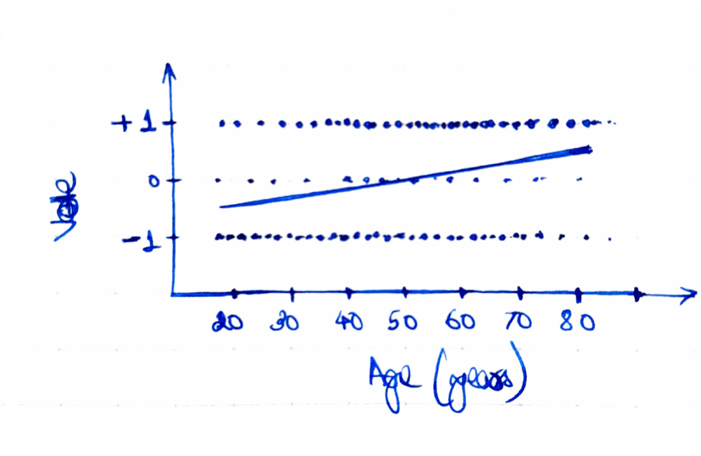


Answer 2

(a) Following the discussion in ROS about regression to the mean dealing with exam scores, we guess that while most pairs (x, y) of scores will follow the line $(x, 2x)$, the regression will yield coefficients of the form $a = 25$, $b = 1$, $\sigma = 10$.

(b) We guess that there is a shift from Democratic to Republican votes as age increases, but with the discrepancy being 60%-40$ at most. The independent vote is estimated to be very little, perhaps simply proportional to the age distribution (any non-zero effect of age here is perhaps too small to illustrate). With this, we guess that our regression line passes through $(20, -0.2)$ and $(80, 0.2)$. Thus, $a = -1/3$, $b = 1/150$, $\sigma = 1$.





Answer 3

(a)

```r
dbinom(3, 10, 0.40)
```

```
## [1] 0.2149908
```

(b)

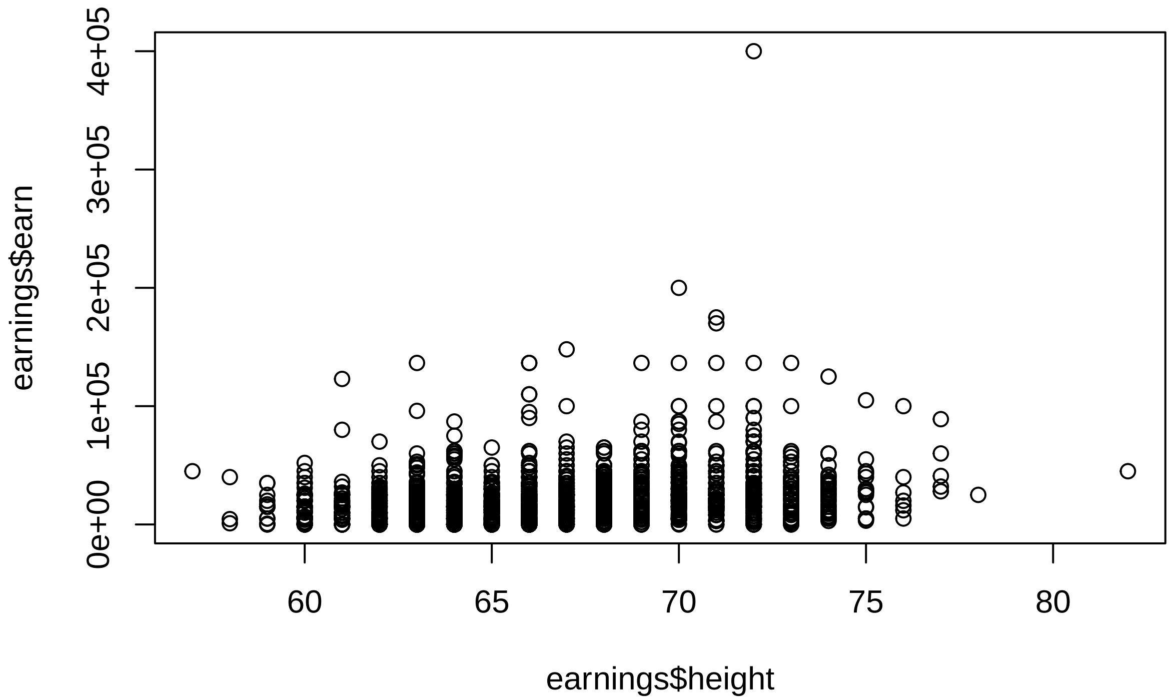
```r
shots <- function(n, p) rbinom(1, n, p)
mean(replicate(10000, shots(10, 0.40) == 3))
```

```
## [1] 0.2158
```

Research homework assignment

We take a first look at our height and earnings data.

```r
earnings <- read.csv("earnings.csv")
plot(earnings$height, earnings$earn)
```

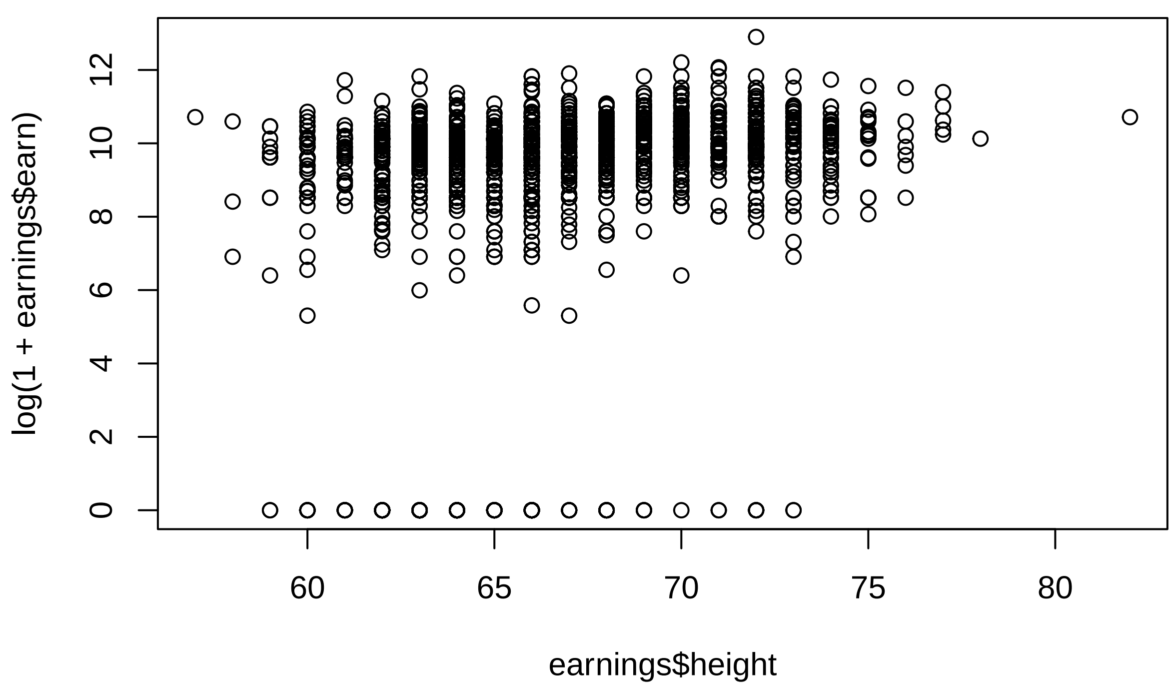


Around 10% of people have 0 earnings. Note that a simple transformation of the form $\log(1+y)$ places these data points in an isolated pile at the bottom.

```
mean(earnings$earn == 0)
```

```
## [1] 0.1029736
```

```
plot(earnings$height, log(1 + earnings$earn))
```



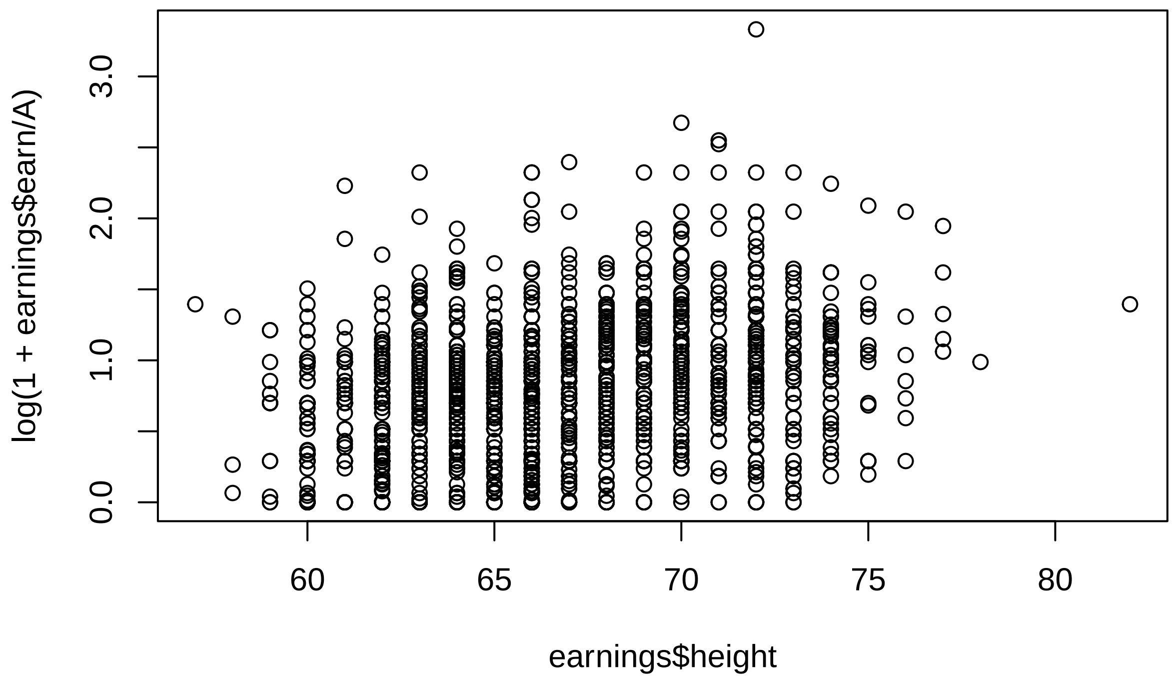
Instead, we look at the transformation $\log(A+y)$. Temporarily, we can also look at $\log(1+y/A)$, which differs from $\log(A+y)$ only by a constant $\log(A)$. This also sends zeroes to zeroes. In this new settings, we may interpret A as a scaling factor (which also helps make y/A dimensionless). With this, we suggest using

$A = \text{MAD}(\mathbf{y})$.

```r
A <- mad(earnings$earn)
print(A)
```

```
## [1] 14826
```

```r
plot(earnings$height, log(1 + earnings$earn/A))
```



We can fit this model as follows. The coefficients can be read off of the `stan_glm` output.

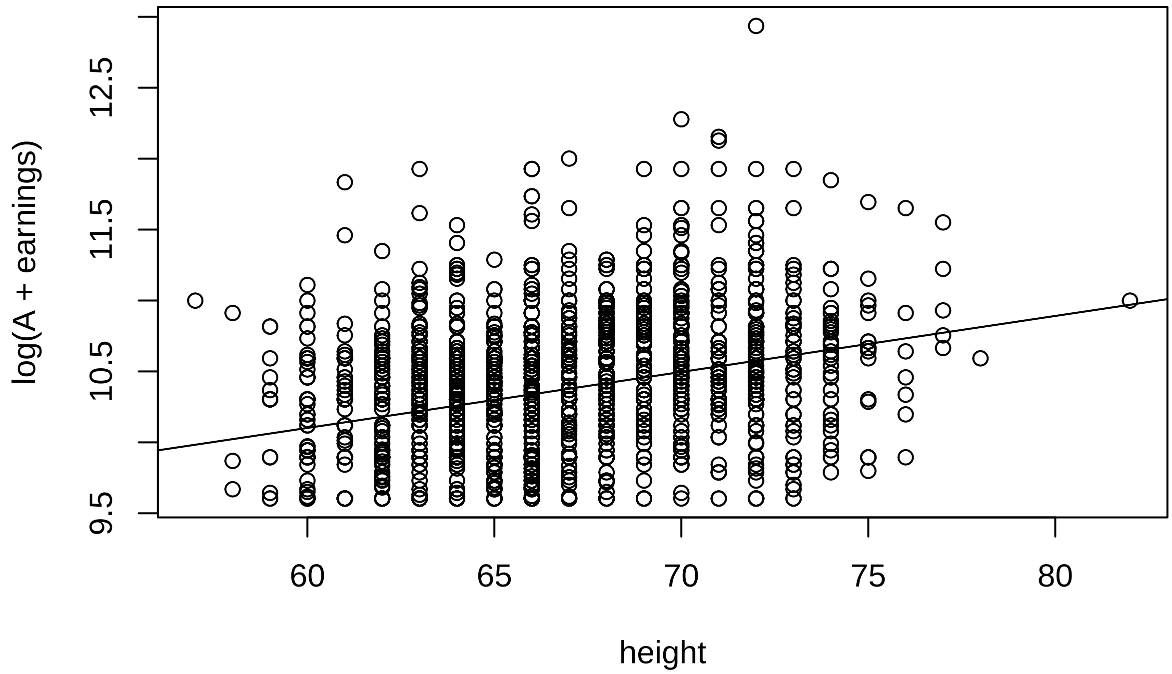
```r
x <- earnings$height
y <- earnings$earn
y.l <- log(A + y)
df <- data.frame(x, y.l)

fit.y.l <- stan_glm(y.l ~ x, data = df, refresh = 0)
print(fit.y.l)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      y.l ~ x
##  observations: 1816
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 7.7    0.2
## x           0.0    0.0
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 0.5    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
```

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

```r
plot(x, y.l, xlab = "height", ylab = "log(A + earnings)")
abline(coef(fit.y.l)[1], coef(fit.y.l)[2])
```



We can also take a look at the residuals, which are approximately normally distributed.

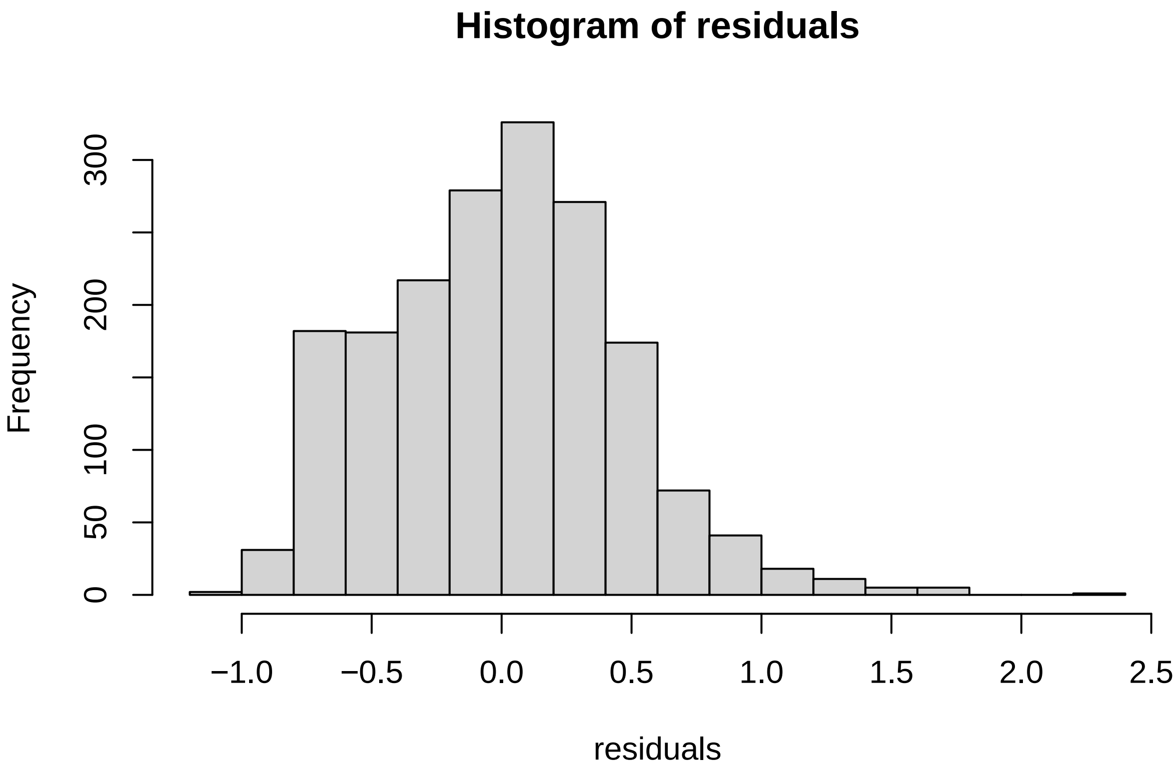
```r
residuals <- y.l - (coef(fit.y.l)[1] + coef(fit.y.l)[2] * x)
shapiro.test(residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.98257, p-value = 4.487e-14
```

```r
R2 <- 1 - sum(residuals^2) / sum((y.l - mean(y.l))^2)
print(R2)
```

```
## [1] 0.09668816
```

```r
hist(residuals, breaks = 16)
```



It is difficult to suggest an appropriate scaling factor A for y without resorting to some measure of dispersion such as the Median Absolute Deviation or standard deviation. Choosing small values of A makes the problem of separation between the $y = 0$ values from the rest of the data worse. Choosing large values of A flattens the bulk of the data far too much and stretches out the top of the plot.