Homework 2: Evaluating Randomized Experiments

Neeraj Sharma

05/02/2020

1. Estimate the treatment effect and the associated standard error on the raw data given to you, assuming there are no mistakes in or problems with the data.
2. It's always a good idea to check a data set for errors. Clean this data set as you think appropriate. As an answer to this question, note all the kinds of changes you made to the data, a few words explaining your reason for the change, and which observations you changed (noting the observation number included as a variable in the data set for identification purposes). If it is totally obvious which observations you changed and there are a large number in the category (e.g. if you decided to drop all White study participants), you can just note what you did for that change ("I dropped all white participants") and explain why.
3. With your cleaned data set, re-estimate the treatment effect and estimated standard error, assuming the randomization worked fine.
4. Evaluate whether the randomization appears legitimate. If you don't think the randomization is legitimate, what is your evidence? (Hint: something went wrong.)
5. Offer your best hypothesis/hypotheses as to what went wrong with the randomization? What evidence do you have to support your hypothesis(es)? For each of these hypotheses, describe your best strategy for estimating a plausible treatment effect, in spite of the bad randomization. (But don't actually estimate that treatment effect.)
6. Given your answer to question five come up with your best estimate of the true treatment effect in the experiment as well as its standard error.

In 1 and 3, assume randomization worked. For the rest, then you can't assume randomization worked. How to estimate if it's random. Multiple linear regression isn't awesome for prediction but we should use it here for randomness assessment.

Multiple Linear Regression Example

fit <- lm(y ~ x1 + x2 + x3, data=mydata) summary(fit) # show results anova function can be used to compare different linear models

Question 1: Estimate the treatment effect and the associated standard error on the raw data given to you, assuming there are no mistakes in or problems with the data.

By definition, the Average Treatment Effect is defined to be $\frac{1}{N}\sum_i y_1(i) - y_0(i)$, but it is often impractical to utilize this approach. This formula presumes one can quantify the outcome of a given individual when treatment is both given and witheld. However, because each individual can only be slotted into one category, this approach cannot be perfectly achieved.

Thus, a random experiment with a control and treatment group can be conducted to smooth out differences amongst populations in order to isolate the treatment effect specifically. With large enough sample groups, the difference between the mean of the outcome of the treatment group and mean of the outcome of the control group yields the Average Treatment Effect.

Here are the relevant summary statistics of the data set without any modification.

Table 1: Summary Statistics of Hours Exercised across Sample Groups with No Data Cleaning

Treatment	Count	Mean	St Dev	St Err
0	500	20.952	84.02863	3.757875
1	500	27.546	104.36699	4.667434

The mean hours exercised for individuals in the treatment group is 27.546 and the mean hours exercised for individuals in the non-treatment group is 20.952. Assuming randomness was properly implimented in this study, the difference between these two numbers will be the Average Treatment Effect of the treatment based on the analysis I provide above. Thus, the Average Treatment Effect is 6.594 with a standard error of 0.9096.

How to tell if you aren't random. Clues that it is the case is done through a balance table. Compare means of each observable metric between treatment and control groups. You do a t-test to compare them. THIS IS NOT GREAT, but still do it to confirm you are balanced on your observables.

```
# Clean the data to impose uniformity upon the variable encoding.
data <- raw %>%
  # Recoding female variable to be factor categorical variable from 0/1/male/female.
  mutate(female = as.double(if_else(female == "female",
                           "1", if_else(female == "male",
                                        "0", female)))) %>%
  # Encode education variable to be a factor
  mutate(education = factor(education, levels = c("less than high school",
                                                  "high school",
                                                  "higher degree"))) %>%
  # Fix all BLACK observations to normal capitalization structure.
  mutate(race_ethnicity = str_to_sentence(race_ethnicity),
         race_ethnicity = factor(race_ethnicity, c("Black", "Hispanic", "White"))) %>%
  # -99 is missing age data so I reincode it at missing age data. https://cran.r-project.org/web/packag
  mutate(age = na_if(age, -99)) %>%
  # Fixing messed up hours readings. They start at 60 and upwards and that's 1 hr so I fix based on tha
  mutate(hours = if_else(hours >= 60, hours / 60, hours)) %>%
  # Currently I have removed improper values, but I could also justify multiplying by 10.
  mutate(bmi = ifelse(bmi < 1, NA, bmi))
```

```
# Checking randomization

# Balance Table
data %>%
  group_by(treatment) %>%
  summarize_all(mean) %>%
  kable()
```

```
## Warning in mean.default(community_center): argument is not numeric or
## logical: returning NA

## Warning in mean.default(community_center): argument is not numeric or
## logical: returning NA

## Warning in mean.default(education): argument is not numeric or logical:
## returning NA

## Warning in mean.default(education): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(race_ethnicity): argument is not numeric or
## logical: returning NA

## Warning in mean.default(race_ethnicity): argument is not numeric or
## logical: returning NA
```

treatment	subject_id	hours	community_center	female	age	bmi	education	race_ethnicity
0	517.356	5.612	NA	0.636	NA	NA	NA	NA
1	483.644	7.486	NA	0.582	NA	NA	NA	NA

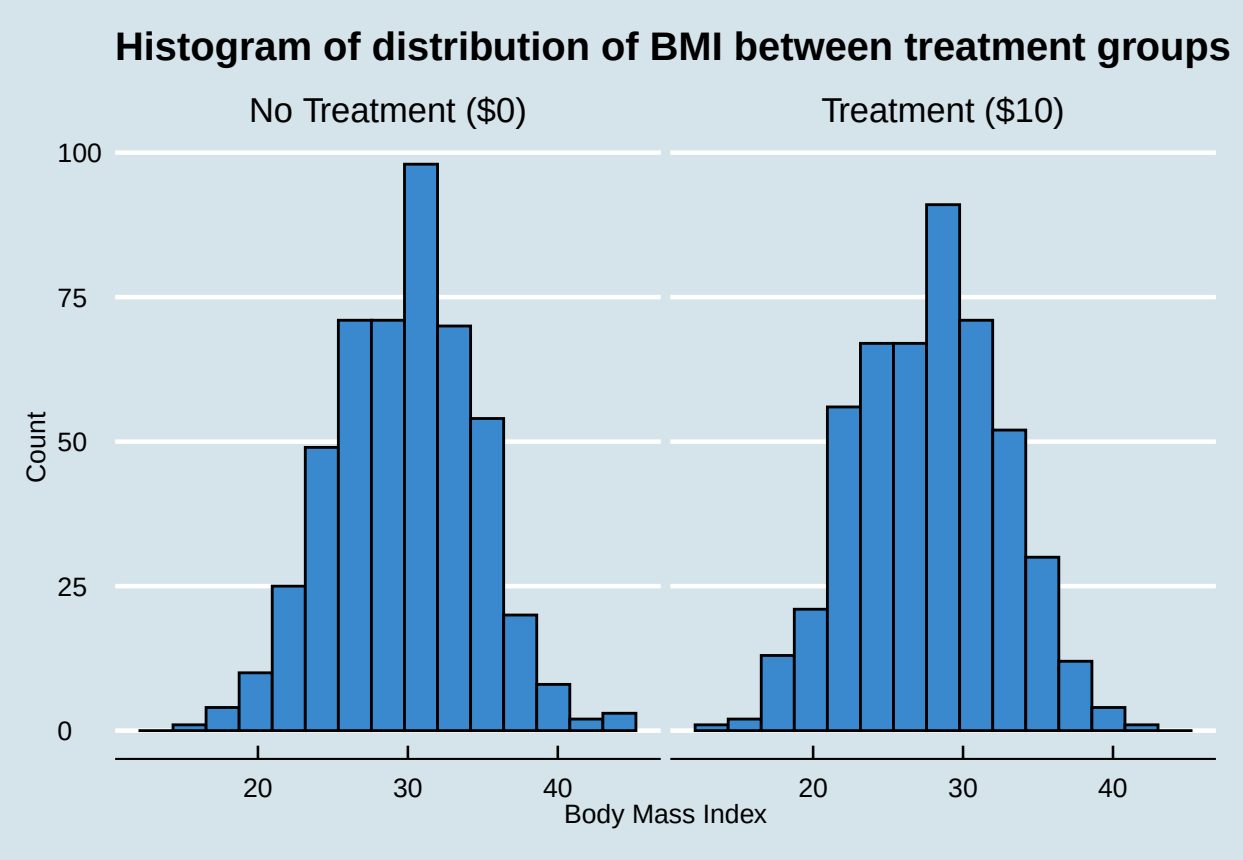
```
data %>%
  select(race_ethnicity) %>%
  count(race_ethnicity) %>%
  kable(col.names = c("Race", "Count"))
```

```
## Warning: Factor `race_ethnicity` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

Race	Count
Black	509
Hispanic	137
White	326
NA	28

```
randomness <- data %>%
  mutate(treatment = factor(treatment, labels = c("No Treatment ($0)", "Treatment ($10)")))

randomness %>%
  drop_na(bmi) %>%
ggplot(mapping = aes(x = bmi)) +
  geom_histogram(fill = "#3a89cf", color = "black", bins = 15) +
  labs(x = "Body Mass Index", y = "Count", title = "Histogram of distribution of BMI between treatment g
  theme_economist() +
  facet_wrap(~ treatment)
```

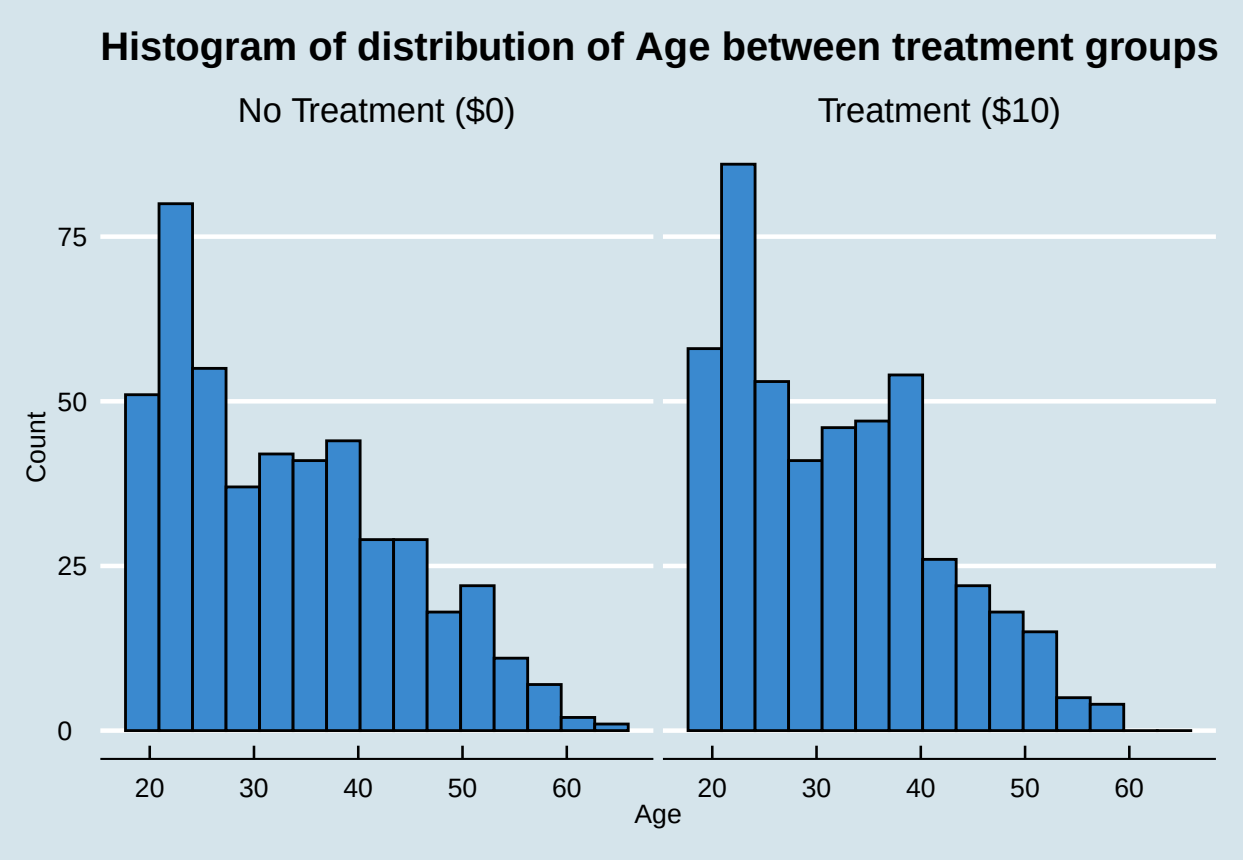



```
randomness %>%
  drop_na(bmi) %>%
  group_by(treatment) %>%
  summarize(count = n(), mean(bmi), sd(bmi), se = sd(bmi)/sqrt(n())) %>%
  kable(col.names = c("Treatment", "Count", "Mean", "St Dev", "St Err"), caption = "Summary Statistics
```

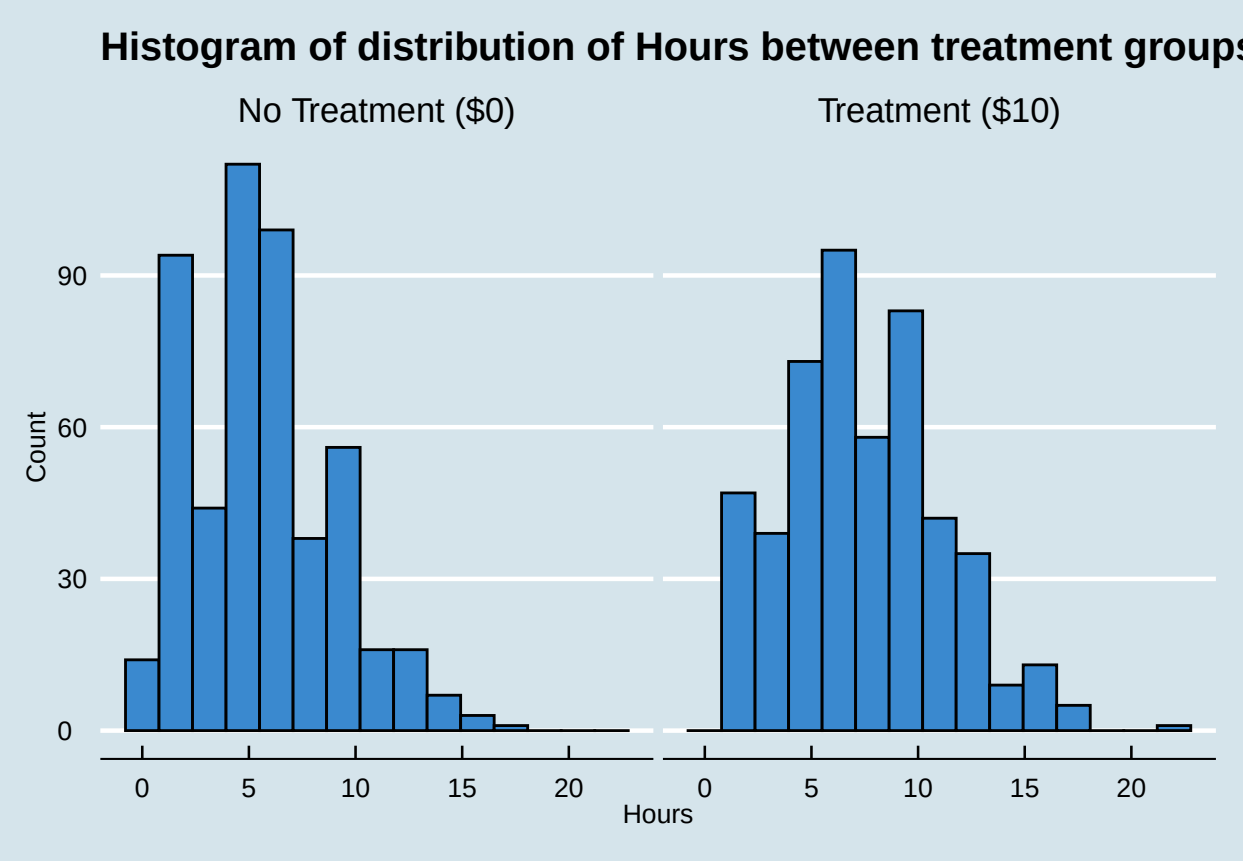
Table 4: Summary Statistics of BMI across Sample Groups

Treatment	Count	Mean	St Dev	St Err
No Treatment ($0)	486	29.78517	4.728381	0.2144837
Treatment ($10)	488	27.78196	4.848491	0.2194809

```
randomness %>%
  drop_na(age) %>%
ggplot(mapping = aes(x = age)) +
  geom_histogram(fill = "#3a89cf", color = "black", bins = 15) +
  labs(x = "Age", y = "Count", title = "Histogram of distribution of Age between treatment groups") +
  theme_economist() +
  facet_wrap(~ treatment)
```




```
randomness %>%
ggplot(mapping = aes(x = hours)) +
  geom_histogram(fill = "#3a89cf", color = "black", bins = 15) +
  labs(x = "Hours", y = "Count", title = "Histogram of distribution of Hours between treatment groups")
  theme_economist() +
  facet_wrap(~ treatment)
```



It is clear that there are numerous outlires in terms of BMI data. Given that a BMI of less than 18.5 is underweight, having a BMI of ~1 is underweight to the point of imposibility. Thus, I believe these data points were improperly encoded and given the distribution, they appear to have simply misplaced the decimal point two places to the left.

##Question 4:Evaluate whether the randomization appears legitimate. If you don't think the randomization is legitimate, what is your evidence? (Hint: something went wrong.)

If there was perfect random assignment, you could randomly pluck 100 people, then pluck 100 more, their distributions should be the same of age BMI Community etc. Systematic differences imply that it's not random. Talk about mean median and stuff but you don't need to do a t-test.

```
treatment_bmi <- glm(treatment ~ bmi, data = data, family = binomial)
summary(treatment_bmi)
```

```
##
## Call:
## glm(formula = treatment ~ bmi, family = binomial, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6614  -1.1313   0.6906   1.1426   1.6819
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.51458    0.40548   6.202 5.59e-10 ***
## bmi         -0.08721    0.01390  -6.275 3.50e-10 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1350.2  on 973  degrees of freedom
## Residual deviance: 1308.5  on 972  degrees of freedom
##   (26 observations deleted due to missingness)
## AIC: 1312.5
##
## Number of Fisher Scoring iterations: 4
```

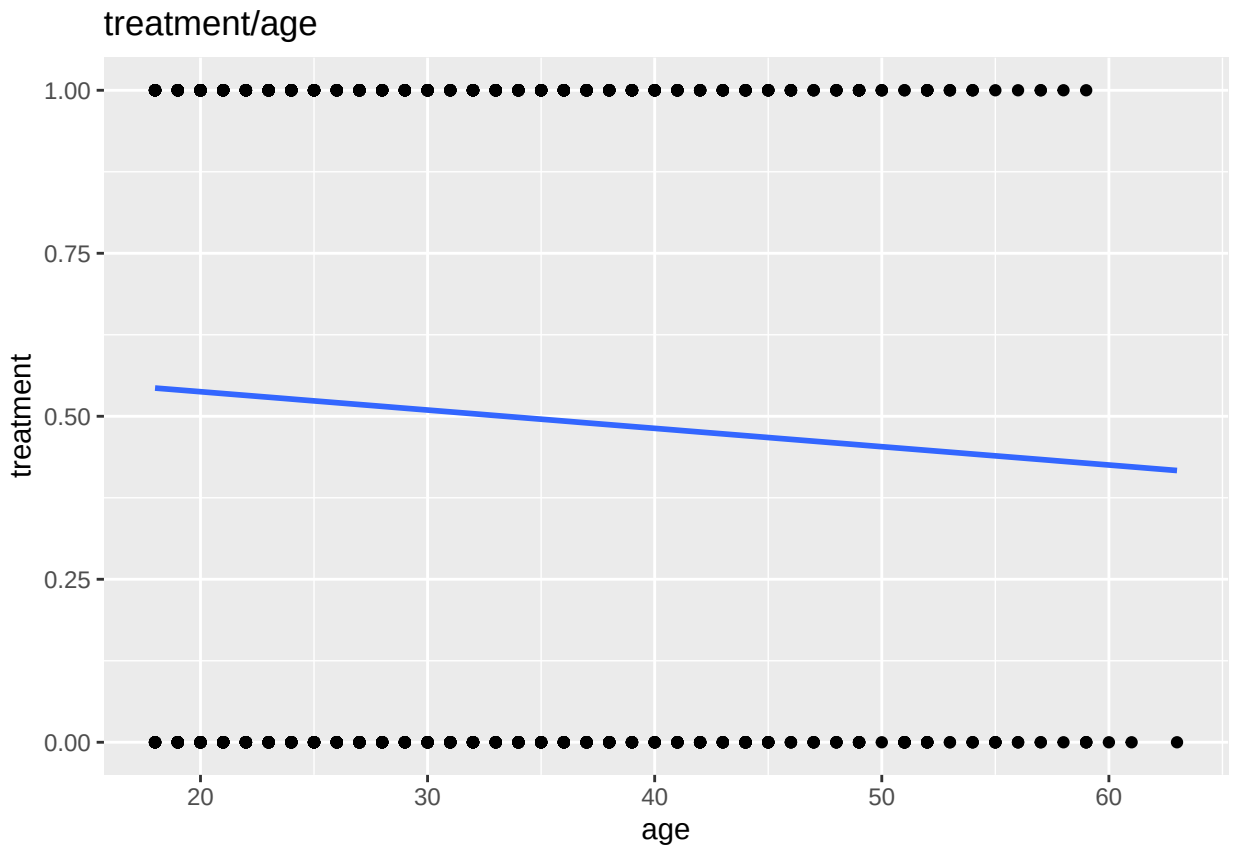
```
test <- broom::augment(treatment_bmi, type.predict = "response") %>%
  mutate(.pred = as.numeric(.fitted > .5))

mean(test$treatment != test$.pred, na.rm = TRUE)
```

```
## [1] 0.4281314
```

```
ggplot(data = data %>% filter(age > 0), mapping = aes(x = age, y = treatment)) + geom_point() +
  geom_smooth(method = "lm", method.args = list(family = "binomial"), se = FALSE) +
  labs(title = "treatment/age")
```

```
## Warning: In lm.wfit(x, y, w, offset = offset, singular.ok = singular.ok,
##     ...) :
##  extra argument 'family' will be disregarded
```

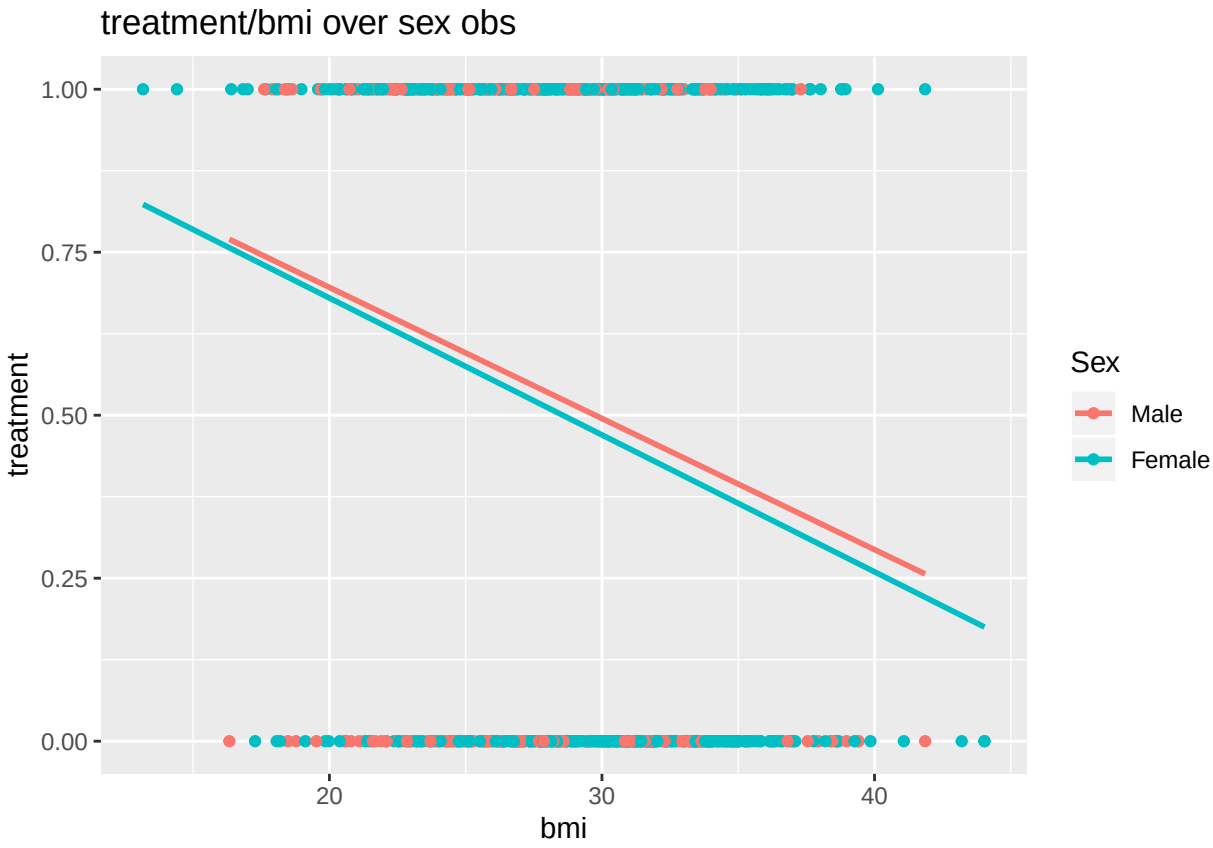



```
data %>%
  filter(bmi > 10, age > 0) %>%
  mutate(female = factor(female, levels = c(0, 1), labels = c("Male", "Female"))) %>%
```

```
ggplot(mapping = aes(x = bmi, y = treatment, color = female)) +
  geom_point() +
  geom_smooth(method = "lm", method.args = list(family = "binomial"), se = FALSE) +
  scale_color_discrete(name = "Sex") +
#  scale_y_discrete(breaks = c(0, 1), limits = c(0, 1)) +
  labs(title = "treatment/bmi over sex obs")
```

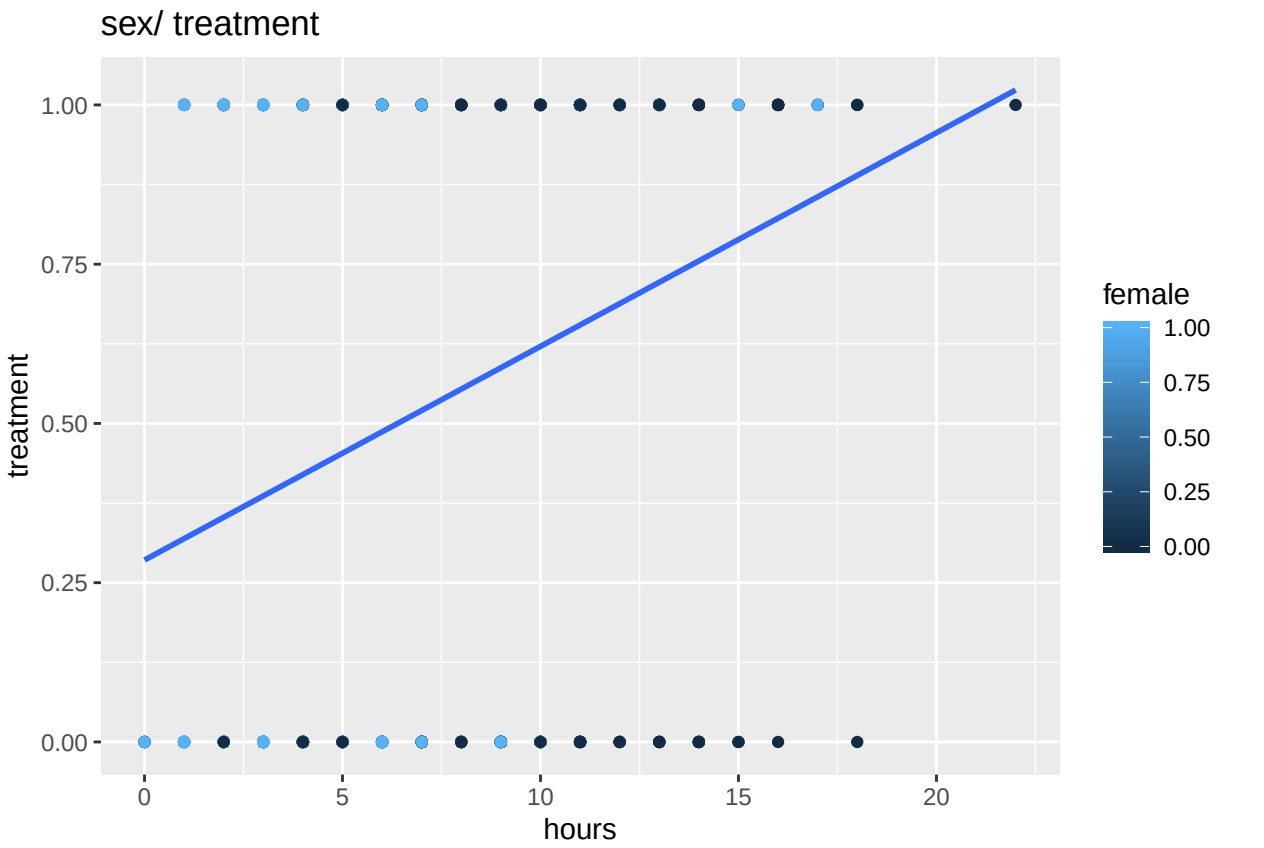
```
## Warning: In lm.wfit(x, y, w, offset = offset, singular.ok = singular.ok,
##     ...) :
##  extra argument 'family' will be disregarded
```

```
## Warning: In lm.wfit(x, y, w, offset = offset, singular.ok = singular.ok,
##     ...) :
##  extra argument 'family' will be disregarded
```



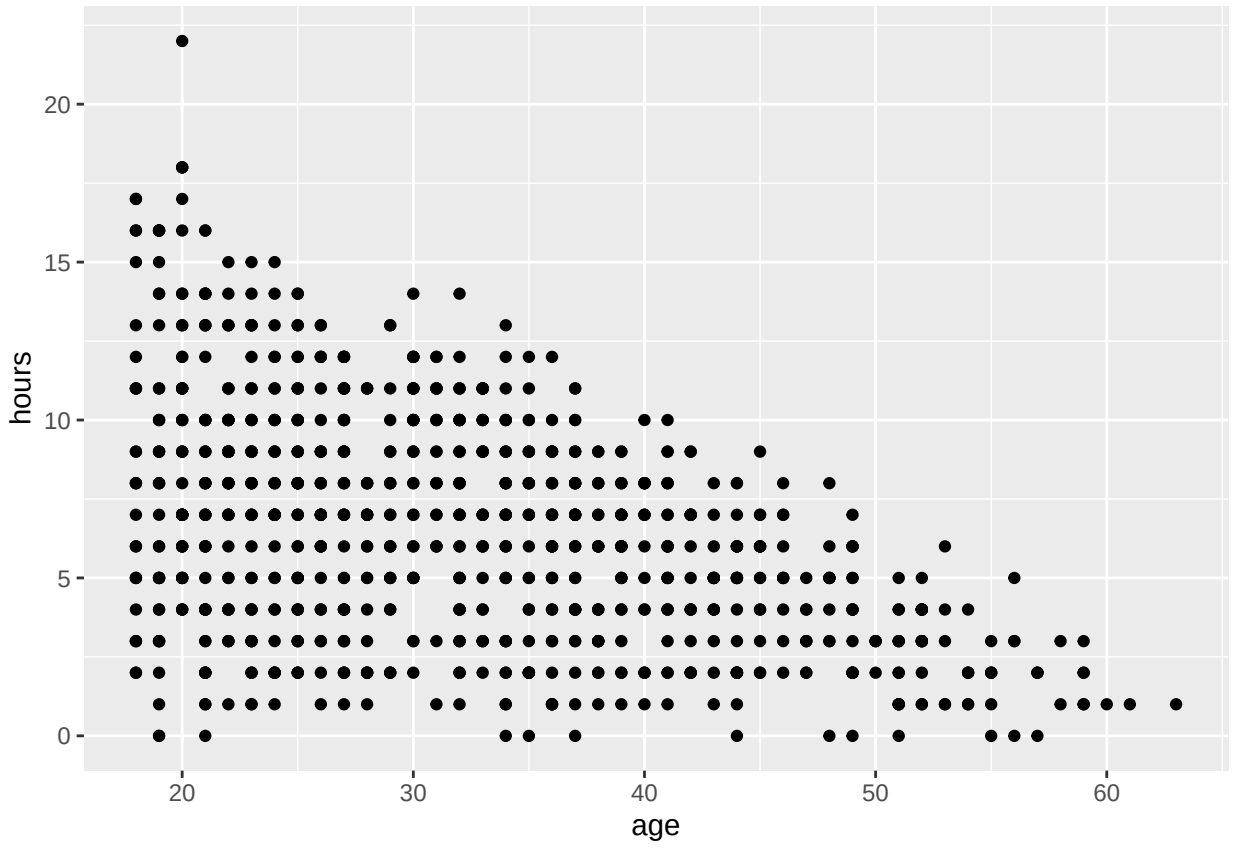

```
ggplot(data = data %>% filter(bmi > 10, age > 0), mapping = aes(x = hours, y = treatment, color = female
  geom_point() +
  geom_smooth(method = "lm", method.args = list(family = "binomial"), se = FALSE) +
  labs(title = "sex/ treatment")
```

```
## Warning: In lm.wfit(x, y, w, offset = offset, singular.ok = singular.ok,
##     ...) :
##  extra argument 'family' will be disregarded
```




```
data %>%
  ggplot(mapping = aes(x = age, y = hours)) +
  geom_point()
```

```
## Warning: Removed 56 rows containing missing values (geom_point).
```


hours
20
15
10
5
0
20
30
40
50
60
age