Weighting, data management and regression

Prof. Dr. Stephan Huber[1,2]

[1] Fresenius University of Applied Science
[2] Charlotte Fresenius University

Abstract

In this document I show what weighted means and their distribution are all about. Furthermore, I show some possibilities of data management in R with the `dplyr` package and how a regression analysis in R is performed and visualised.

All files related to this document can be found here: https://github.com/hubchev/ewa. Please contact us via `stephan.huber@hs-fresenius.de`.

Table 1
Data

v1	v2	v3	v4
1	A	NA	
2	NA	0.2	0.4
	C	0.3	0.1
4	D	NA	3
5	E	0.5	1.5

1 Solutions and Cheatsheet

Please consider the report you find here:

https://hubchev.github.io/various/exam_functions.html

In this report, I summarize operators and popular functions of R. Moreover, I present the output of all exercises. That should help you to write code and start to look for solutions to your challenges in working with data:

2 Datenmanagement

Consider the data of Table 1 and solve the following exercises:

a) Add variable `misone` that is 1 if there is a missing and 0 otherwise. *(Hint: Use `case_when` and `is.na()`.)*

Table 2
Solution a)

v1	v2	v3	v4	misone
1	A	NA		1
2	NA	0.2	0.4	1
	C	0.3	0.1	1
4	D	NA	3	1
5	E	0.5	1.5	0

```
df <- df |>
  mutate(misone = case_when(
    v1 == "" | is.na(v1) ~ 1,
    v2 == "" | is.na(v2) ~ 1,
    v3 == "" | is.na(v3) ~ 1,
    v4 == "" | is.na(v4) ~ 1,
    TRUE ~ 0
  ))

knitr::kable(df, "latex", caption = "Solution a)")
```

b) Add variable `miscount` that counts how many observations are missing in each row. *(Hint: Use `mutate_all`, `rowSums`, and `pick(everything())`)*

Table 3
Solution b)

v1	v2	v3	v4	misone	miscount
1	A	NA		1	2
2	NA	0.2	0.4	1	1
	C	0.3	0.1	1	1
4	D	NA	3	1	1
5	E	0.5	1.5	0	0

```
test_df <- df |>
  mutate_all(~ if_else(is.na(.) | . == "", 1, 0)) |>
  mutate(miscount = rowSums(pick(everything())))

test_df_miscount <- test_df |>
  select(miscount)

df <- bind_cols(df, test_df_miscount)

knitr::kable(df, "latex", caption = "Solution b)")
```

c) Use the function `rowwise` to calculate the `NA` and `""` observations. *(Hint: Use* ***`is.na`*** *and* ***`pick(everything())`****.)*

Table 4
Solution c)

v1	v2	v3	v4	misone	miscount	count_NA	count_OK
1	A	NA		1	2	1	1
2	NA	0.2	0.4	1	1	1	0
	C	0.3	0.1	1	1	0	1
4	D	NA	3	1	1	1	0
5	E	0.5	1.5	0	0	0	0

```
df <- df |>
  rowwise() |>
  mutate(count_NA = sum(is.na(pick(everything())))) |>
  mutate(count_OK = sum(pick(everything()) == "", na.rm = TRUE)) |>
  ungroup()

knitr::kable(df, "latex", caption = "Solution c)")
```

d) Add variable `mispercent` that measures the percentage of missings and a variable mis30up that is 1 if the percentage is above 30%. *(Hint: Use* `mutate`*,* `select`*,* `ifelse`*, and* `bind_cols`*.)*

Table 5
Solution d)

v1	v2	v3	v4	misone	miscount	count_NA	count_OK	mis30up	fraction
1	A	NA		1	2	1	1	1	0.50
2	NA	0.2	0.4	1	1	1	0	0	0.25
	C	0.3	0.1	1	1	0	1	0	0.25
4	D	NA	3	1	1	1	0	0	0.25
5	E	0.5	1.5	0	0	0	0	0	0.00

```
test_df_mis30up <- test_df |>
  mutate(fraction = miscount / 4) |>
  mutate(mis30up = ifelse(fraction > 0.3, 1, 0)) |>
  select(mis30up, fraction)

df <- bind_cols(df, test_df_mis30up)

knitr::kable(df, "latex", caption = "Solution d)")
```

e) Calculate the average of the numeric variables `v1`, `v3`, and `v4`. Name the variable `average`. *(Hint: Use* `as.numeric`*,* `rowwise`*, and* `mean`*.)*

Table 6
Solution e)

v1	v3	v4	average
1	NA	NA	1.0000000
2	0.2	0.4	0.8666667
NA	0.3	0.1	0.2000000
4	NA	3.0	3.5000000
5	0.5	1.5	2.3333333

```
df <- df |>
  mutate(
    v1 = as.numeric(v1),
    v4 = as.numeric(v4)
  )
df <- df |>
  rowwise() |>
  mutate(average = mean(c(v1, v3, v4), na.rm = TRUE)) |>
  ungroup()

test_df <- df |>
  select(v1, v3, v4, average)

knitr::kable(test_df, "latex", caption = "Solution e)")
```

3 Regression

Please consider my lecture notes concerning **Regression Analysis** which you find here:

https://hubchev.github.io/qm/statistics.html#simple-linear-regression

Moreover, I highly recommend reading Wysocki et al. (2022) which is freely available here: https://journals.sagepub.com/doi/10.1177/25152459221095823. They explain how difficult it is to use regression analysis to dentify a causal impact. The main insights of the paper are nicely summarized here: https://osf.io/38mxq.

3.1 Making regression tables using `apa_table`

Here is an example how to use `apa_table` from the `papaja` package to make regression output tables.

```
# Load the mtcars dataset
data("mtcars")

# Fit a linear regression model
m1 <- lm(mpg ~ wt + hp, data = mtcars)
m2 <- lm(mpg ~ wt , data = mtcars)

# Summary of the model
summary(m1)
```

```
##
## Call:
## lm(formula = mpg ~ wt + hp, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.941 -1.600 -0.182  1.050  5.854
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.22727    1.59879  23.285  < 2e-16 ***
## wt          -3.87783    0.63273  -6.129 1.12e-06 ***
## hp          -0.03177    0.00903  -3.519  0.00145 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.593 on 29 degrees of freedom
## Multiple R-squared:  0.8268, Adjusted R-squared:  0.8148
## F-statistic: 69.21 on 2 and 29 DF,  p-value: 9.109e-12
```

Table 7
A full regression table.

Predictor	*b*	95% CI	*t*	*df*	*p*
Intercept	37.23	[33.96, 40.50]	23.28	29	$< .001$
Wt	-3.88	[-5.17, -2.58]	-6.13	29	$< .001$
Hp	-0.03	[-0.05, -0.01]	-3.52	29	.001

```
apa_lm <- apa_print(m1)
apa_table(
  apa_lm$table
  , caption = "A full regression table."
)
```

4 Example

4.1 Data

In the statistic course of WS 2020, I asked 23 students about their weight, height, sex, and number of siblings:

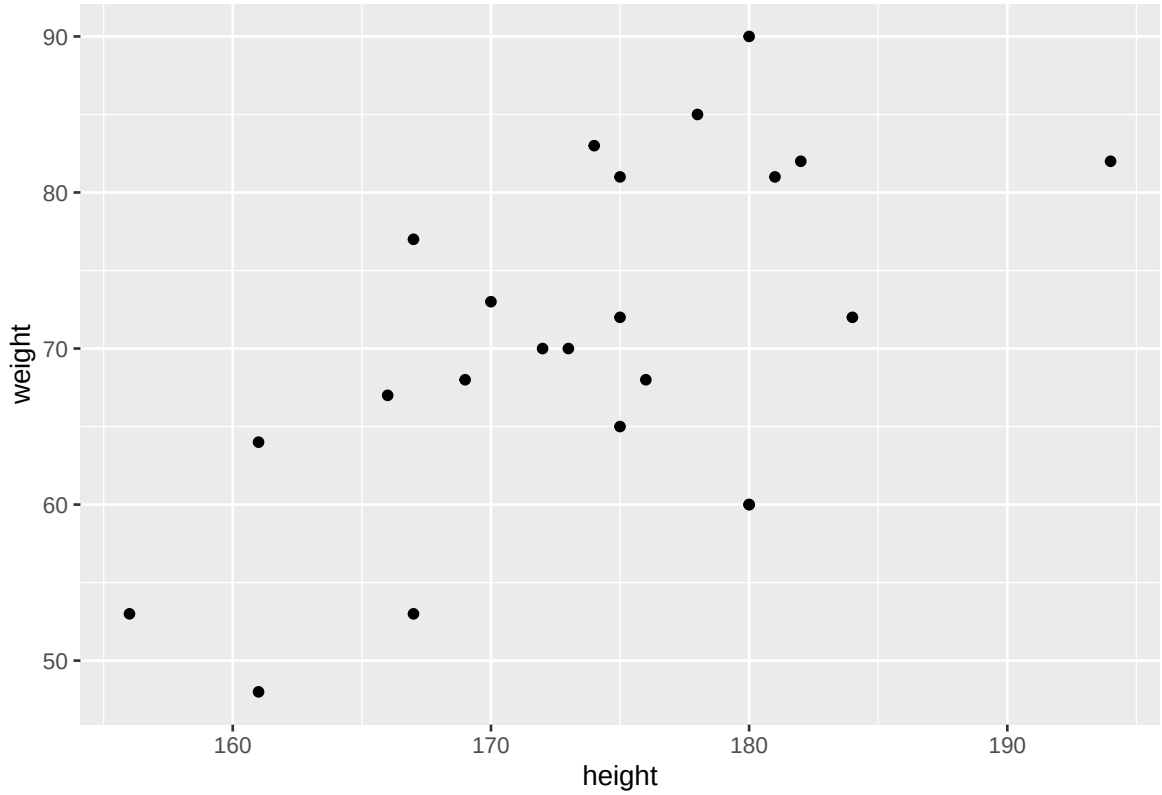
```
library("haven")
classdata <- read.csv("https://raw.githubusercontent.com/hubchev/courses/main/dta/classdata

head(classdata)
```

```
##   id sex weight height siblings row
## 1  1   w     53    156        1   g
## 2  2   w     73    170        1   g
## 3  3   m     68    169        1   g
## 4  4   w     67    166        1   g
## 5  5   w     65    175        1   g
## 6  6   w     48    161        0   g
```

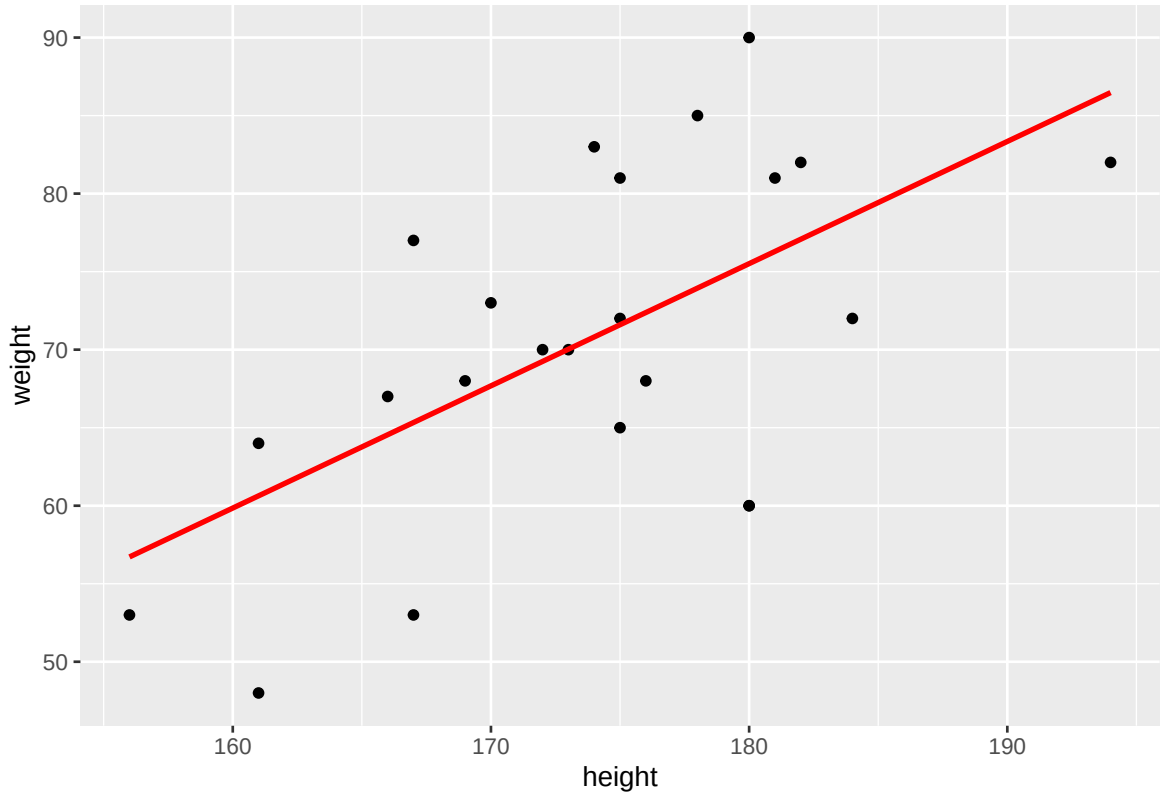
4.2 First look at data

```
library("ggplot2")
ggplot(classdata, aes(x=height, y=weight)) + geom_point()
```



4.3 Include a regression line:

```
ggplot(classdata, aes(x=height, y=weight)) +
  geom_point() +
  stat_smooth(formula=y~x, method="lm", se=FALSE, colour="red", linetype=1)
```



4.4 Regression: Distinguish male/female by including a seperate constant:

```
## baseline regression  model
model  <- lm(weight ~ height + sex , data = classdata )
show(model)
```

```
##
## Call:
## lm(formula = weight ~ height + sex, data = classdata)
##
## Coefficients:
## (Intercept)       height         sexw
##    -29.5297       0.5923      -5.7894
```

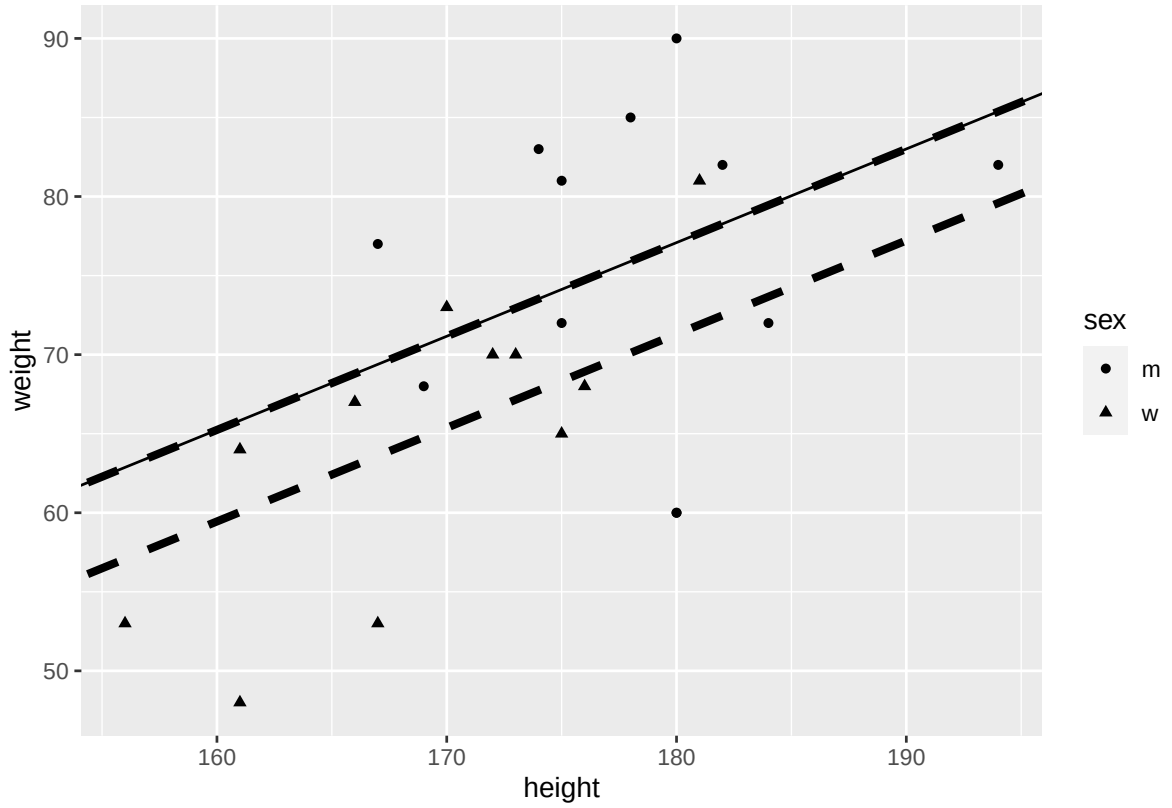
```
interm <- model$coefficients[1]
slope  <- model$coefficients[2]
interw <- model$coefficients[1]+model$coefficients[3]
```

```
summary(model)
```

```
##
## Call:
## lm(formula = weight ~ height + sex, data = classdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.086  -3.730   2.850   7.245  12.914
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -29.5297    47.6606  -0.620   0.5425
## height        0.5923     0.2671   2.217   0.0383 *
## sexw         -5.7894     4.4773  -1.293   0.2107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.942 on 20 degrees of freedom
## Multiple R-squared:  0.4124, Adjusted R-squared:  0.3537
## F-statistic: 7.019 on 2 and 20 DF,  p-value: 0.004904
```

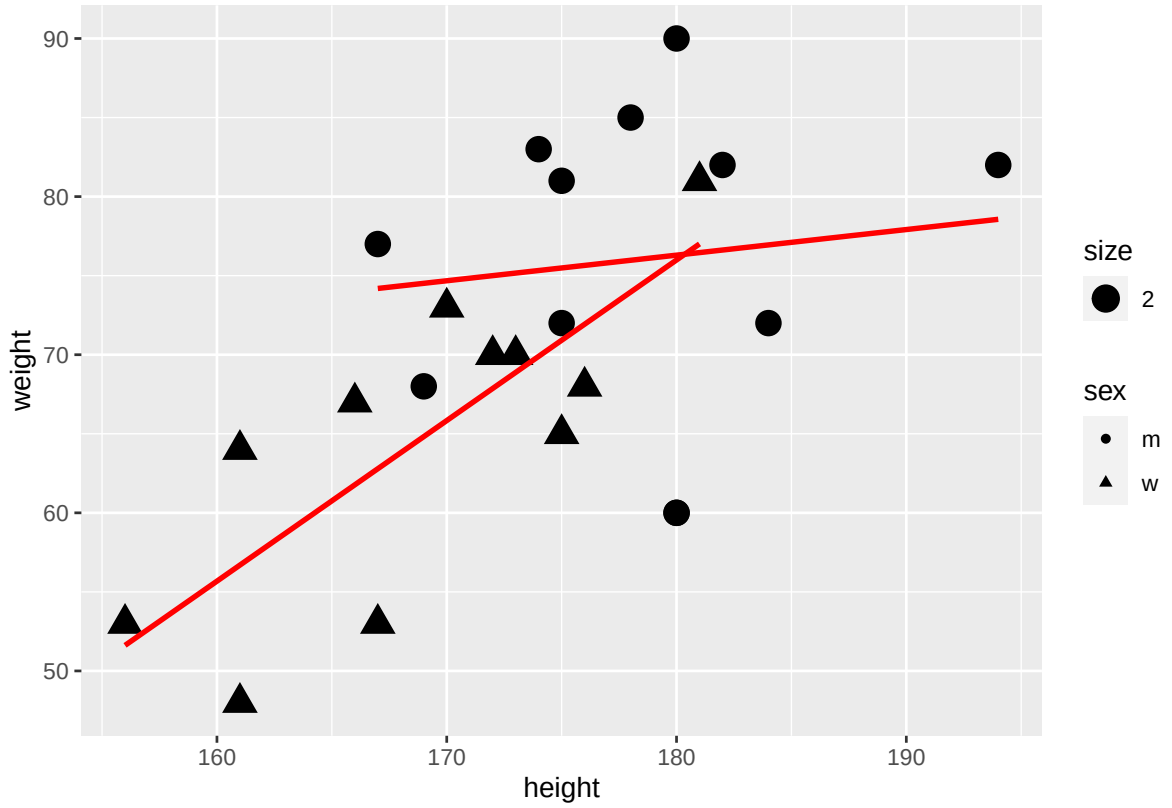
```
ggplot(classdata, aes(x=height, y=weight, shape = sex)) +
  geom_point() +
  geom_abline(slope = slope, intercept = interw, linetype = 2, size=1.5)+
  geom_abline(slope = slope, intercept = interm, linetype = 2, size=1.5) +
  geom_abline(slope = coef(model)[[2]], intercept = coef(model)[[1]])
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```



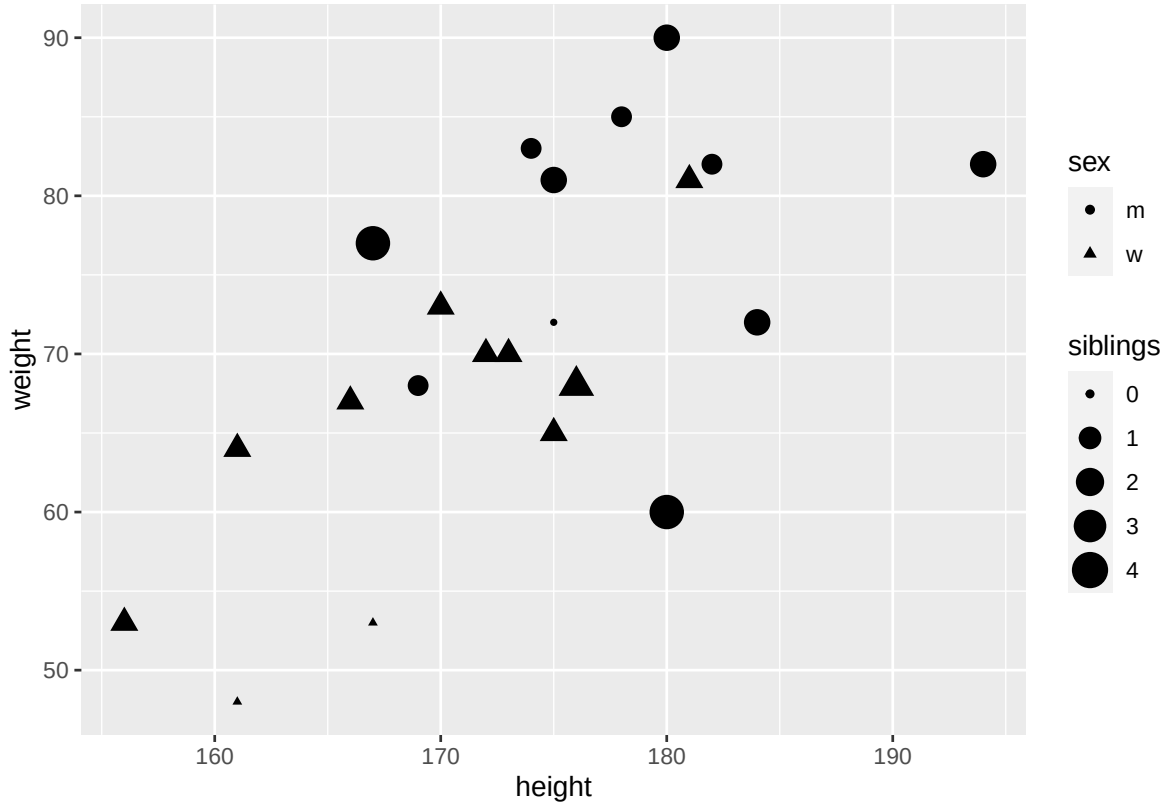
That does not look good. Maybe we should introduce also different slopes for male and female.

```
ggplot(classdata, aes(x=height, y=weight, shape = sex)) +
  geom_point( aes(size = 2)) +
  stat_smooth(formula = y ~ x,  method = "lm",
              se = FALSE, colour = "red", linetype = 1)
```



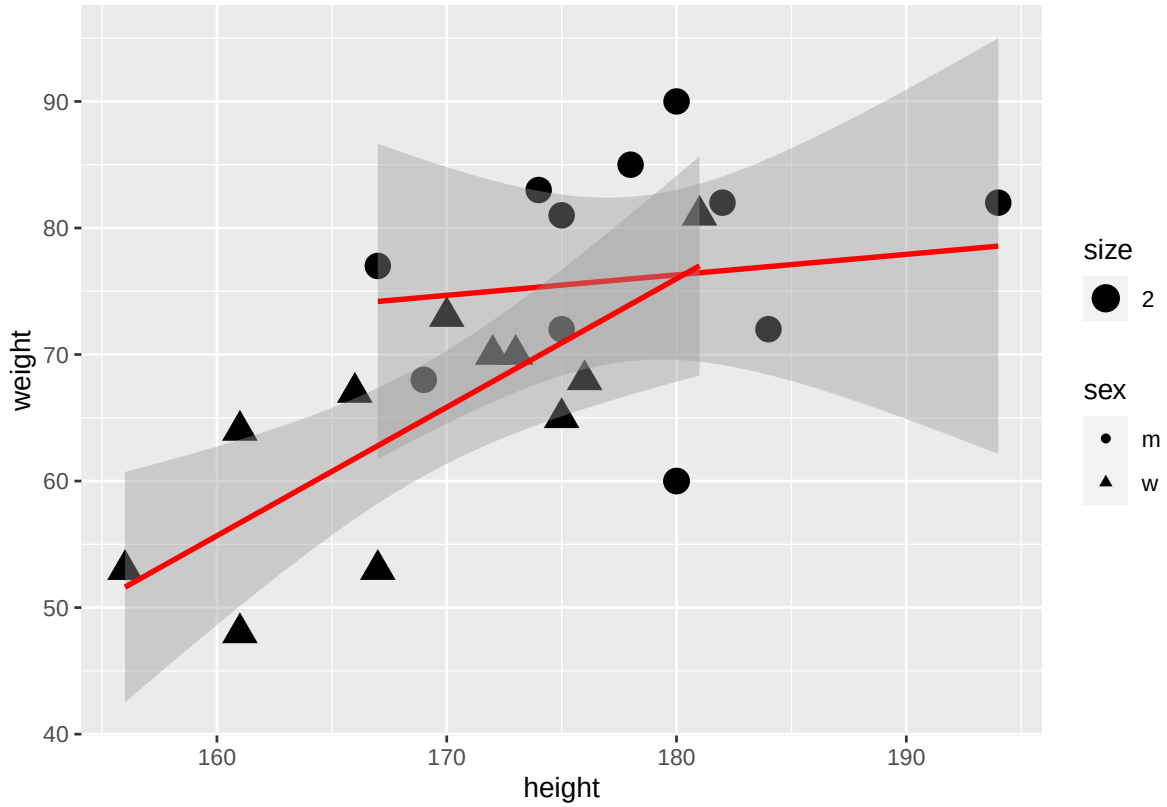
4.5 Can we use other available variables such as siblings?

```
ggplot(classdata, aes(x=height, y=weight, shape = sex)) +
  geom_point( aes(size = siblings))
```




```
## baseline model
model  <- lm(weight ~ height + sex , data = classdata )

ggplot(classdata, aes(x=height, y=weight, shape = sex)) +
  geom_point( aes(size = 2)) +
  stat_smooth(formula = y ~ x,
              method = "lm",
              se = T,
              colour = "red",
              linetype = 1)
```


90
80
70
60
50
40
weight
160
170
180
190
height
size
2
sex
m
w

4.6 Let us look at regression output:

```
m1 <- lm(weight ~ height , data = classdata )
m2 <- lm(weight ~ height + sex , data = classdata )
m3 <- lm(weight ~ height + sex + height * sex , data = classdata )
m4 <- lm(weight ~ height + sex + height * sex + siblings , data = classdata )
m5 <- lm(weight ~ height + sex + height * sex , data = subset(classdata, siblings < 4 ))
```

Table 8
Regression

	Dependent variable:				
			weight		
	Model-1	Model-2	Model-3	Model-4	Model-5
	(1)	(2)	(3)	(4)	(5)
height	0.78***	0.59**	0.16	0.16	0.28
	(0.23)	(0.27)	(0.36)	(0.37)	(0.39)
sexw		−5.79	−153.96*	−161.92*	−134.51
		(4.48)	(88.96)	(91.68)	(90.65)
siblings				−1.16	
				(2.05)	
height:sexw			0.85	0.89	0.74
			(0.51)	(0.53)	(0.52)
Constant	−65.44	−29.53	47.14	50.27	27.69
	(39.35)	(47.66)	(64.81)	(66.23)	(70.36)
Observations	23	23	23	23	21
R^2	0.36	0.41	0.49	0.50	0.57
Adjusted R^2	0.33	0.35	0.41	0.38	0.50
Residual Std. Error	9.08	8.94	8.57	8.73	8.04
F Statistic	11.98***	7.02***	6.02***	4.44**	7.59***
Note:	*p<0.1; **p<0.05; ***p<0.01				
	Here are my notes.				

4.7 Interpretation of the results

- We can make predictions about the impact of height on male and female
- As both, the intercept and the slope differs for male and female we should interpret the regressions seperately:
- One centimeter more for **MEN** is *on average* and *ceteris paribus* related with 0.16 kg more weight.
- One centimeter more for **WOMEN** is *on average* and *ceteris paribus* related with 1.01 kg more weight.

4.8 Regression Diagnostics

Linear Regression makes several assumptions about the data, the model assumes that:

- The relationship between the predictor (x) and the dependent variable (y) has linear relationship.
- The residuals are assumed to have a constant variance.
- The residual errors are assumed to be normally distributed.
- Error terms are independent and have zero mean.

More on regression Diagnostics can be found Applied Statistics with R: 13 Model Diagnostics

5 Weighting

The formula for the weighted mean is:

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i \cdot x_i}{\sum_{i=1}^{n} w_i}$$

In this formula:

- $\bar{x}$ represents the weighted mean.
- n is the number of observations.
- w_i represents the weight for the i-th observation.
- x_i represents the i-th observation value.

```
rm(list = ls())
wt <- c(5, 2, 2, 1)
x <- c(1, 2, 3, 4)
x_mean <- mean(x)
x_mean
```

```
## [1] 2.5
```

```
x_wt_mean_1 <- weighted.mean(x, wt)
x_wt_mean_1
```

```
## [1] 1.9
```

Let us calculate the weighted mean manually:

```
product <- wt*x
# Nominator
nom <- sum(product)
nom
```

```
## [1] 19
```

```
# Denominator
denom <- sum(wt)
denom
```

```
## [1] 10
```

```
x_wt_mean_2 <- nom/denom
x_wt_mean_2
```

```
## [1] 1.9
```

5.1 Exercise 1

Below you see an alternative way to calculate the weighted mean. Can you explain it?

```
w_div_sumw <- wt/denom
w_div_sumw
```

```
## [1] 0.5 0.2 0.2 0.1
```

```
multi_ww_x <- w_div_sumw * x
multi_ww_x
```

```
## [1] 0.5 0.4 0.6 0.4
```

```
x_wt_mean_3 <- sum(multi_ww_x)
x_wt_mean_3
```

```
## [1] 1.9
```

5.2 Exercise 2

a) Calculate mean, variance, weighted mean, and the variance of the weighted mean for x.

```
results <- data.frame(
  Statistic = c("Mean", "Variance", "Weighted Mean", "Weighted Variance"),
  Value = c(mean(x), var(x), weighted.mean(x, wt), sum(wt * (x - weighted.mean(x, wt))^2) /
)
print(results)
```

```
##           Statistic    Value
## 1              Mean 2.500000
## 2          Variance 1.666667
## 3     Weighted Mean 1.900000
## 4 Weighted Variance 1.090000
```

b) Do it again but use `tidyverse` and the function `summarize`.

```
df <- tibble(wt = wt, x = x)

summary_stats <- df %>%
  summarize(
    Mean = mean(x),
    Variance = var(x),
    Weighted_Mean = weighted.mean(x, wt),
    Weighted_Variance = sum(wt * (x - weighted.mean(x, wt))^2) / sum(wt)
  )

# Display the table
print(summary_stats)
```

```
## # A tibble: 1 x 4
##    Mean Variance Weighted_Mean Weighted_Variance
##   <dbl>    <dbl>         <dbl>              <dbl>
## 1   2.5     1.67           1.9               1.09
```

References

Wysocki, A. C., Lawson, K. M., & Rhemtulla, M. (2022). Statistical control requires causal justification. *Advances in Methods and Practices in Psychological Science*, *5*(2). https://doi.org/10.1177/25152459221095823