Solutions to exercises

Listed below are the solutions to the exercises.

All solutions are found using RStudio, though **you should only do the exercises in RStudio if indicated in the list of exercises.** This may result in slight differences in numerical answers, which is due to rounding errors.

The solutions may often be computed in different ways and when two solutions are given it does not necessarily mean that more solutions does not exist. However, when two solutions are given we encourage you to think about why these two solutions are equivalent.

```
library(mosaic)
```

Module 3.2

11.1:

a.i)

```
hs_GPA <- 4.0
vcbs <- 800
0.20 + 0.50*hs_GPA + 0.002*vcbs
```

```
## [1] 3.8
```

a.ii)

```
hs_GPA <- 3.0
vcbs <- 300
0.20 + 0.50*hs_GPA + 0.002*vcbs
```

```
## [1] 2.3
```

b)

$$\begin{aligned} & 0.20 + 0.50x_1 + 0.002 * 500 \\ = \; & 0.20 + 0.50x_1 + 1 \\ = \; & 1.20 + 0.50x_1 \end{aligned}$$

c)

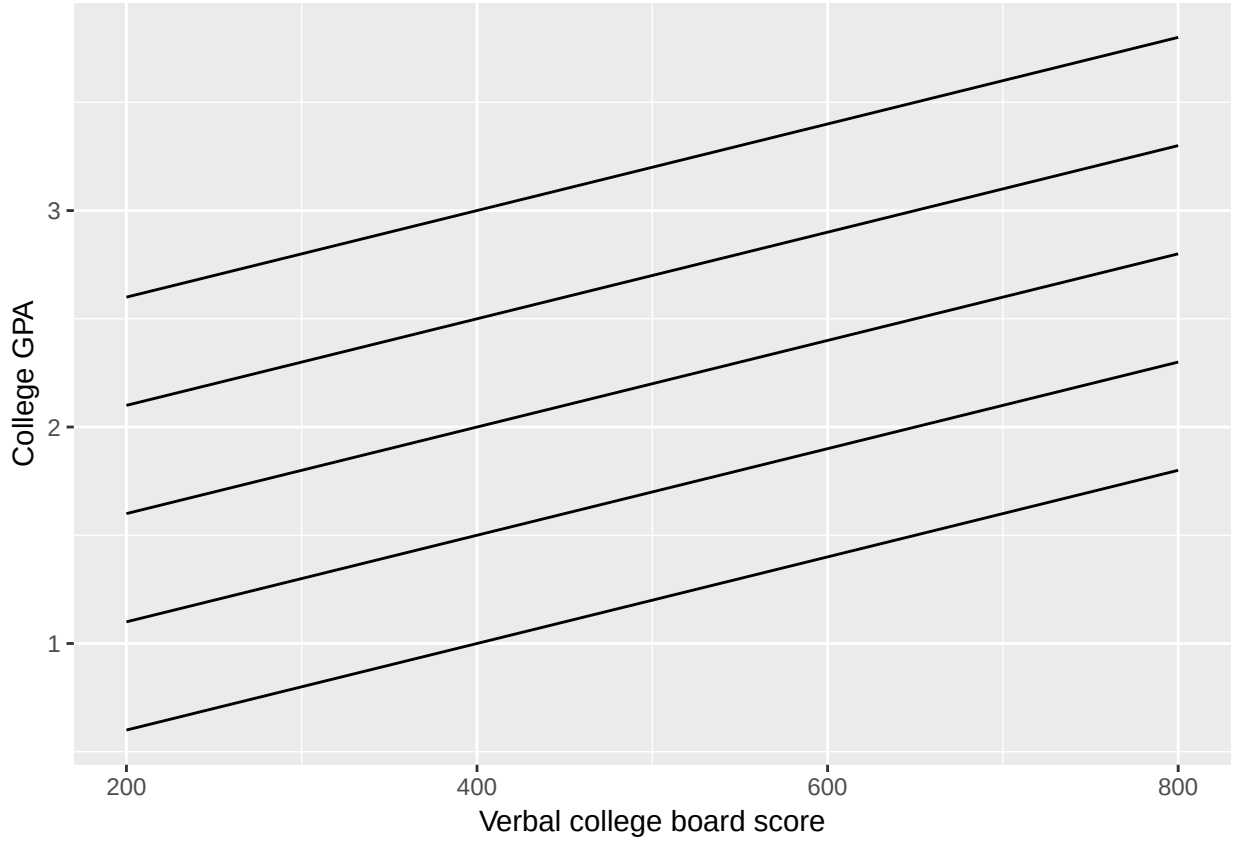
$$\begin{aligned} & 0.20 + 0.50x_1 + 0.002 * 600 \\ = \; & 0.20 + 0.50x_1 + 0.002 * 500 + 0.002 * 100 \\ = \; & 0.20 + 0.50x_1 + 1 + 0.2 \\ = \; & 1.40 + 0.50x_1 \end{aligned}$$

d)

By the previous exercise we see that fixating an explanetory variable changes the intercept and the line prediction equation drops one dimension (i.e. from a plane to a line). That is, in this case by fixating high school GPA the prediction equation becomes the equation for a line with slope 0.002, and some intercept that depends on the high school GPA.

```
pred_eq <- function(hs_GPA) {
  function(vcbs) 0.20 + 0.50*hs_GPA + 0.002*vcbs
}

gf_function(fun = pred_eq(hs_GPA = 0.0), xlim = c(200, 800)) %>%
  gf_function(fun = pred_eq(hs_GPA = 1.0)) %>%
  gf_function(fun = pred_eq(hs_GPA = 2.0)) %>%
  gf_function(fun = pred_eq(hs_GPA = 3.0)) %>%
  gf_function(fun = pred_eq(hs_GPA = 4.0)) %>%
  gf_labs(x = "Verbal college board score", y = "College GPA")
```



11.5:

a)

$$\hat{y} = -3.601 + 1.2799x_1 + 0.1021x_2$$

b)

```r
GDP <- 10
CELLULAR <- 50
-3.601 + 1.2799*GDP + 0.1021*CELLULAR
```

```
## [1] 14.303
```

c.i)

$$\hat{y} = -3.601 + 1.2799\text{GDP}$$

c.ii)

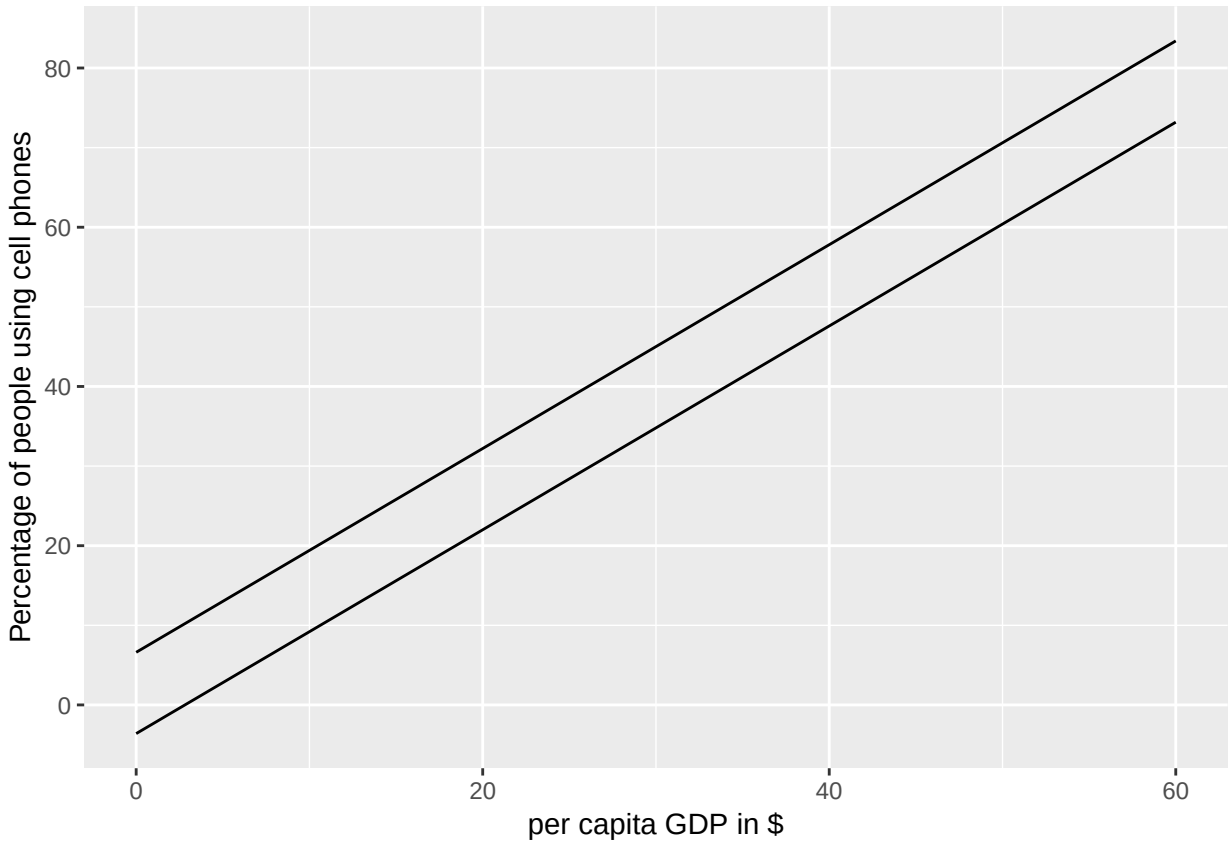
$$\hat{y} = 6.609 + 1.2799\text{GDP}$$

For a fixed value of cell phone usage the the expected number of people using the internet incease by 1.2799 percentage points.

d)

If we plot the two equations we will realise that they are paralelle (as stated in exercise 11.1.d), hence the effect of cell phone usage on the percentage of people using the internet does not depend on the per capital GPA.

```r
pred_eq <- function(CELLULAR) {
  function(GDP) -3.601 + 1.2799*GDP + 0.1021*CELLULAR
}
gf_function(fun = pred_eq(CELLULAR = 0), xlim = c(0, 60)) %>%
  gf_function(fun = pred_eq(CELLULAR = 100)) %>%
  gf_labs(x = "per capita GDP in $", y = "Percentage of people using cell phones")
```



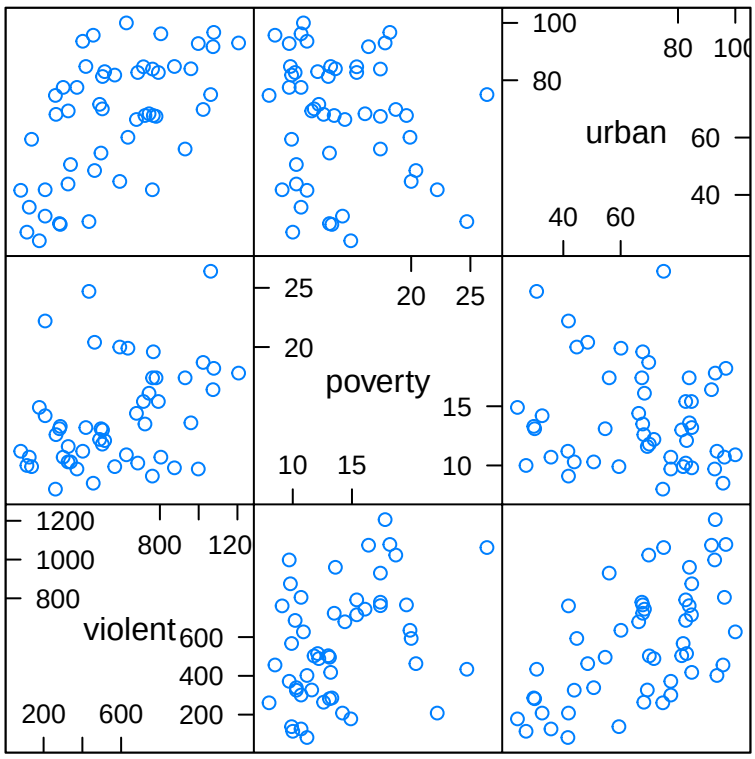
11.11:

Additional subexercises: Import data (also available for download at the website):

```
crime <- read.table("https://asta.math.aau.dk/datasets?file=Crime2.dat", header = TRUE)
crime <- crime[crime$State != "DC", ]
```

Start by making relevant plot(s).

```
splom( ~ crime[c("violent", "poverty", "urban")])
```



Then fit the linear model in R:

```
fit <- lm(violent ~ poverty + urban, data = crime)
summary(fit)
```

```
##
## Call:
## lm(formula = violent ~ poverty + urban, data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -398.41 -150.38    2.89   96.47  581.94
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -498.683    140.988  -3.537 0.000922 ***
## poverty       32.622      6.677   4.885 1.24e-05 ***
## urban          9.112      1.321   6.900 1.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 197.9 on 47 degrees of freedom
## Multiple R-squared:  0.5708, Adjusted R-squared:  0.5525
## F-statistic: 31.25 on 2 and 47 DF,  p-value: 2.335e-09
```

a)

$$\hat{y} = -498.683 + 32.622x_{\text{poverty}} + 9.112x_{\text{urban}}$$

b)

```
pred <- -498.683 + 32.622 * 10.7 + 9.112 * 96.2
pred
```

```
## [1] 726.9468
```

```
res <- 805 - pred
res
```

```
## [1] 78.0532
```

The residual is positive and thus the actual violent crime rate in Massachusetts is higher than what we expect.

c.i)

$$\hat{y} = -498.683 + 32.622 x_{\text{poverty}}$$

c.ii)

$$\hat{y} = 412.517 + 32.622 x_{\text{poverty}}$$

When fixating x_{urban} we see that the expected change in the violent crime rate per unit increase in poverty is 32.622. Additionally, we see that with a poverty of 0% the crime rate is negative, which is not really sensible. Thus we need to take care when interpreting this kind of model.

d)

```
cor( ~ crime[c("violent", "poverty", "urban")])
```

```
##           violent    poverty      urban
## violent 1.0000000  0.3687547  0.5939627
## poverty 0.3687547  1.0000000 -0.1556215
## urban   0.5939627 -0.1556215  1.0000000
```

The correlation between the violent crime rate and poverty seems to be be weak and lower than the correlation between the violent crime rate and percentage living in urban areas which is moderate, while there is a weak correlation between poverty and the percentage living in urban areas.

e)

```
R_sqr <- summary(fit)$r.squared
R_sqr
```

```
## [1] 0.570765
```

```
sqrt(R_sqr)
```

```
## [1] 0.7554899
```

The multiple correlation is a measure of linear dependance between observed and predicted values. It is 0.76 which is a moderately strong correlation thus making the model moderately good at prediction.

The R^2 value is 0.57 which means that 57% of the variation in the violent crime rate is eplained by the model, which too must be moderately good.

11.21:

Additional subexercises:

```
fit <- lm(violent ~ urban * poverty, data = crime)
summary(fit)
```

```
##
## Call:
## lm(formula = violent ~ urban * poverty, data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -288.00 -118.78  -22.59   99.60  500.84
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   158.8620   289.9498   0.548    0.586
## urban          -1.2852     4.2583  -0.302    0.764
## poverty       -14.7245    19.5849  -0.752    0.456
## urban:poverty   0.7598     0.2975   2.554    0.014 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 187.2 on 46 degrees of freedom
## Multiple R-squared:  0.6241, Adjusted R-squared:  0.5996
## F-statistic: 25.45 on 3 and 46 DF,  p-value: 7.424e-10
```

The interaction is significant but the main effects are not. However, we may not reduce the model because of the hieracical principle stating that we must always test interactions before the corresponding main effects, otherwise the model is not interpretable.

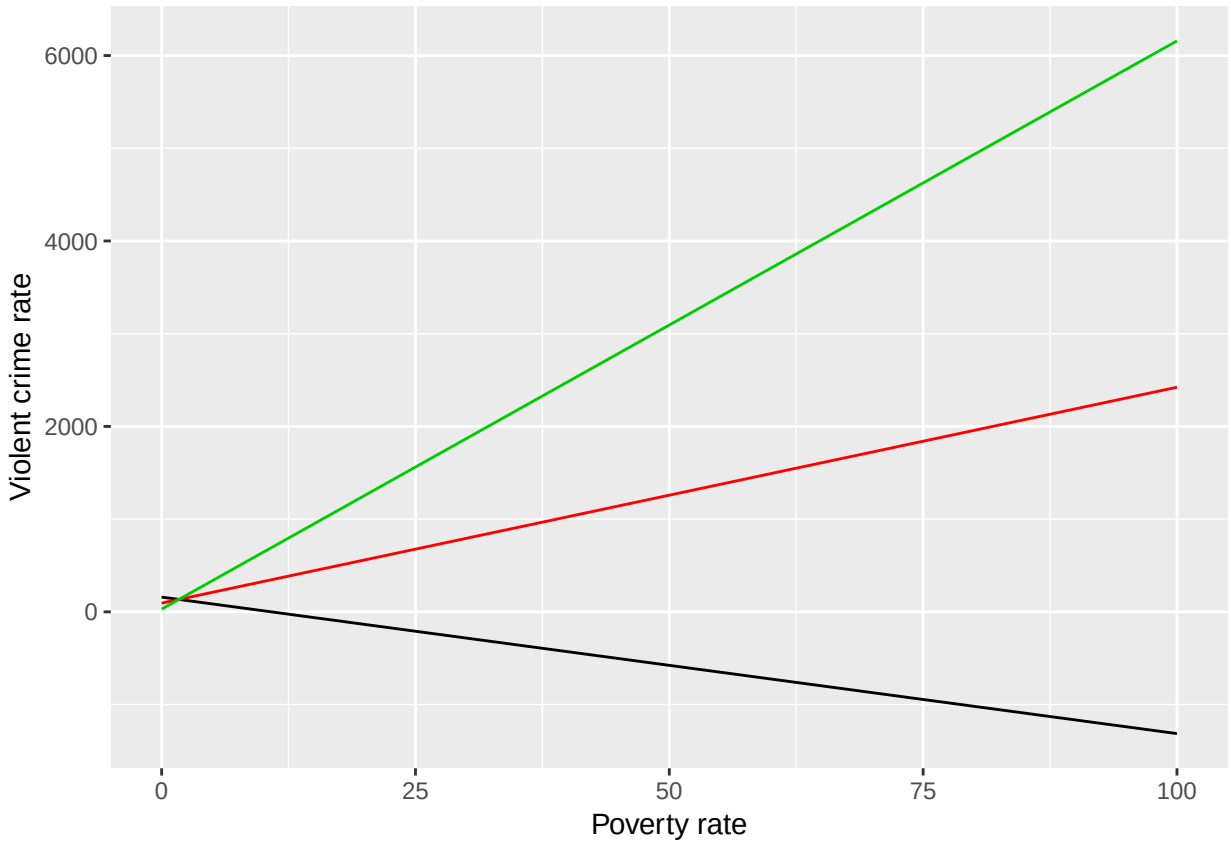
a)

Increase. The estimated parameter of the interaction term is positive.

b)

Plot of the prediction equation for the three fixed values of urban population (0 is black, 50 is red, and 100 is green):

```
pred_eq <- function(x2) {
  function(x1) 158.9 - 14.72 * x1 - 1.29 * x2 + 0.76 * x1 * x2
}
gf_function(fun = pred_eq(0), xlim = c(0, 100)) %>%
  gf_function(fun = pred_eq(50), col = 2) %>%
  gf_function(fun = pred_eq(100), col = 3) %>%
  gf_labs(y = "Violent crime rate", x = "Poverty rate")
```



There is a clear indication of an interaction and we further see that as urban population increases the effect of poverty rate inceases aswell which confirms the previous subexercise. The prediction equations for the values of urban population are as follows:

$$\hat{y} = 158.9 - 14.72x_{\text{Poverty}}$$
$$\hat{y} = 94.4 + 23.28x_{\text{Poverty}}$$
$$\hat{y} = 29.9 + 61.28x_{\text{Poverty}}$$

We observe that as urban population increases the slope (effect) of poverty becomes larger.