Exercise 1

1.

For a fixed value of IQ and GPA,males earn more on average than females.
Depends on fixed value of $X1$, because being female increases $X3$ to 1, which means that the salary changes by $\hat{\beta}_3 + \hat{\beta}_5 X_1 = 35 - 10X_1$.
So false, for a GPA < 3.5 and true for a GPA > 3.5.

For a fixed value of IQ and GPA, females earn more on average than males.
See above.

For a fixed value of IQ and GPA,males earn more on average than females provided that the GPA is high enough.
True, if GPA > 3.5.

For a fixed value of IQ and GPA, females earn more on average than males provided that the GPA is high enough.
False.

2.

$\hat{y} = 50 + 4.0 * 20 + 110 * 0.07 + 1 * 35 + 4.0 * 110 * 0.01 + 4.0 * 1 * (-10) = 137.1$

3.

False, because the effect that the GPA/IQ interaction term has on the salary not only depends on the magnitude of $\hat{\beta}_3$, but also on the magnitude of X_1 and X_2. As IQ is generally around 100, the effect of the GPA/IQ interaction term can be quite high. Also to statistacally measure the effect we would have to look at the p value.

Exercise 2

First we import the relevant libraries and the data.

```
library(MASS) # For Boston data set
library(tidymodels)
# library(ISLR)
library(GGally)
library(broom)
library(dotwhisker)
# library(performance)
# library(funModeling)
```

```
rm( list=ls())
set.seed( 42 )
options(scipen=10000)
```

```
data(Boston)
head(Boston)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
```

```
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

Then we setup up our linear regression specification.

```r
lm_spec <- linear_reg() %>%
  set_mode("regression") %>%
  set_engine("lm")

show_engines("linear_reg")
```

```
## # A tibble: 7 x 2
##   engine mode
##   <chr>  <chr>
## 1 lm     regression
## 2 glm    regression
## 3 glmnet regression
## 4 stan   regression
## 5 spark  regression
## 6 keras  regression
## 7 brulee regression
```

Now we run one linear regression on the crime rate for each predictor.

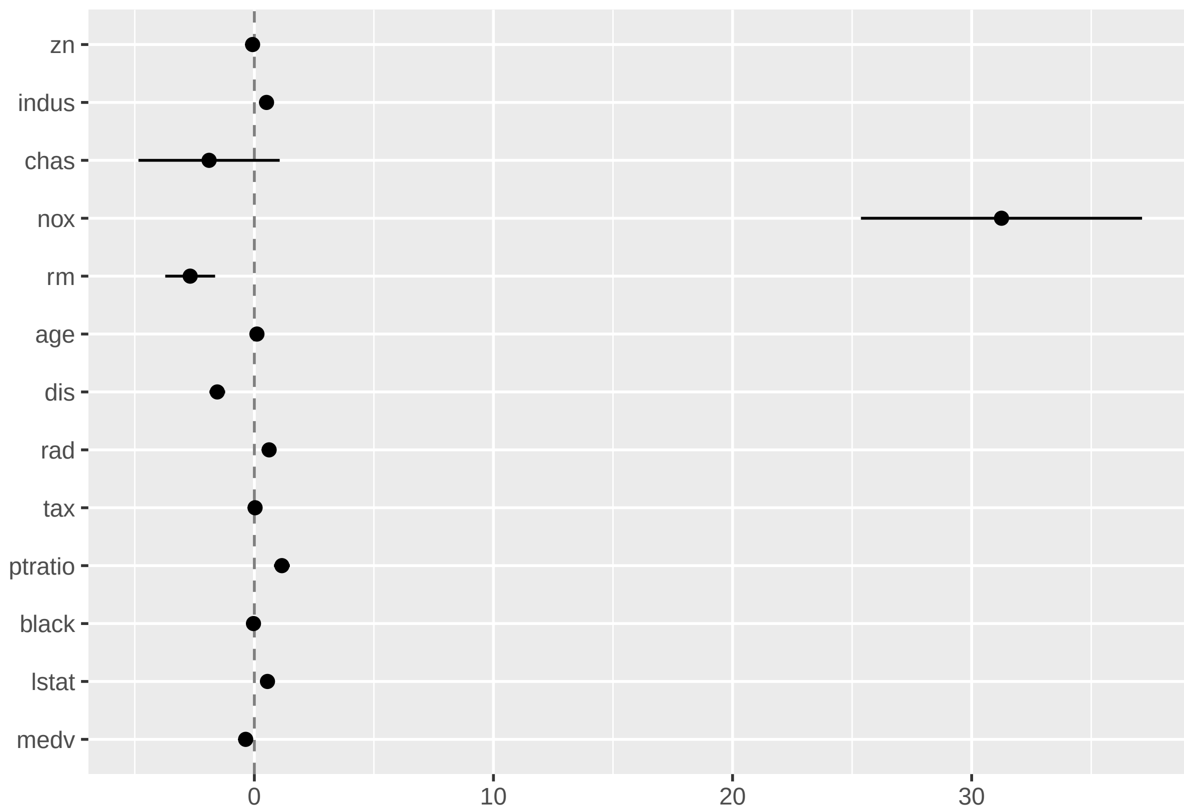
```r
target <- "crim"
# init empty results df
results <- data.frame()
# loop over each column in the data set except the target
for (col in colnames(Boston)) {
  if (col == target) {
      next
  }
  lm_fit <- lm_spec %>%
    fit_xy(
      x = Boston %>% select(all_of(col)),
      y = Boston %>% select(all_of(target))
    )
  term <- tidy(lm_fit)[2, ]
  # append to df
  results <- rbind(results, term)
}
results
```

```
## # A tibble: 13 x 5
##    term    estimate std.error statistic  p.value
##    <chr>      <dbl>     <dbl>     <dbl>    <dbl>
##  1 zn       -0.0739   0.0161      -4.59 5.51e- 6
##  2 indus     0.510    0.0510       9.99 1.45e-21
##  3 chas     -1.89     1.51        -1.26 2.09e- 1
##  4 nox      31.2      3.00        10.4  3.75e-23
##  5 rm       -2.68     0.532       -5.04 6.35e- 7
```

```
##   6 age       0.108    0.0127       8.46 2.85e-16
##   7 dis      -1.55     0.168       -9.21 8.52e-19
##   8 rad       0.618    0.0343      18.0  2.69e-56
##   9 tax       0.0297   0.00185     16.1  2.36e-47
##  10 ptratio   1.15     0.169        6.80 2.94e-11
##  11 black    -0.0363   0.00387     -9.37 2.49e-19
##  12 lstat     0.549    0.0478      11.5  2.65e-27
##  13 medv     -0.363    0.0384      -9.46 1.17e-19
```

Now let's visualize the results with whiskers.

```r
results %>%
  dwplot(dot_args = list(size = 2, color = "black"),
         whisker_args = list(color = "black"),
         vline = geom_vline(xintercept = 0, colour = "grey50", linetype = 2))
```



As we can see the coefficient of the variable nox, which stands for "nitrogen oxides concentration", has the highest magnitude and is significant.

Let us remove it so that we can see the other coefficients better.

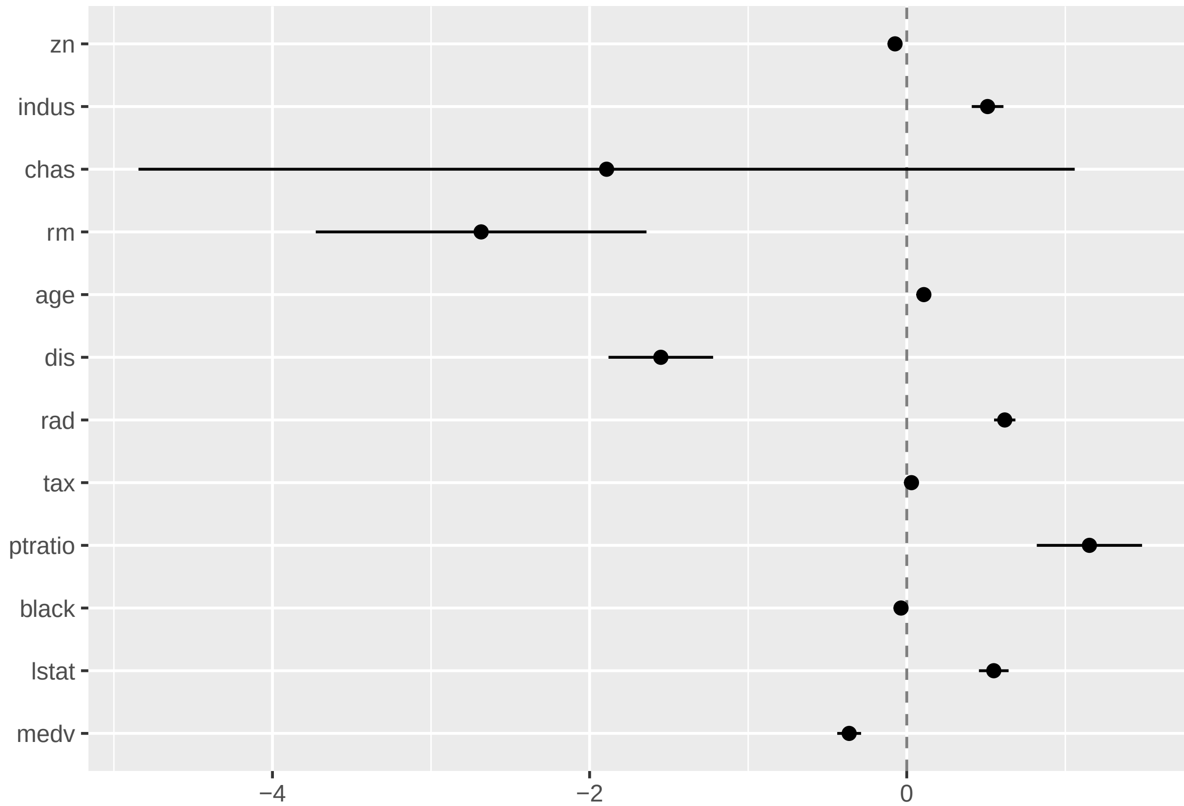
```r
results_sm <- results %>%
  filter(term != "nox")

results_sm
```

```
## # A tibble: 12 x 5
##    term    estimate std.error statistic  p.value
##    <chr>      <dbl>     <dbl>     <dbl>    <dbl>
##  1 zn       -0.0739    0.0161     -4.59 5.51e- 6
##  2 indus     0.510     0.0510      9.99 1.45e-21
```

```
##  3 chas     -1.89     1.51        -1.26 2.09e- 1
##  4 rm       -2.68     0.532       -5.04 6.35e- 7
##  5 age       0.108    0.0127       8.46 2.85e-16
##  6 dis      -1.55     0.168       -9.21 8.52e-19
##  7 rad       0.618    0.0343      18.0  2.69e-56
##  8 tax       0.0297   0.00185     16.1  2.36e-47
##  9 ptratio   1.15     0.169        6.80 2.94e-11
## 10 black    -0.0363   0.00387     -9.37 2.49e-19
## 11 lstat     0.549    0.0478      11.5  2.65e-27
## 12 medv     -0.363    0.0384      -9.46 1.17e-19
```

```r
results_sm %>%
  dwplot(dot_args = list(size = 2, color = "black"),
         whisker_args = list(color = "black"),
         vline = geom_vline(xintercept = 0, colour = "grey50", linetype = 2))
```



Now we see that every predictor has a significant coefficient except chas, which is a dummy variable that tells us whether the Charles River runs through this neighborhood.

However, we will get a we get a better picture of which predictors are relevant if we run a linear regression with all predictors at once.

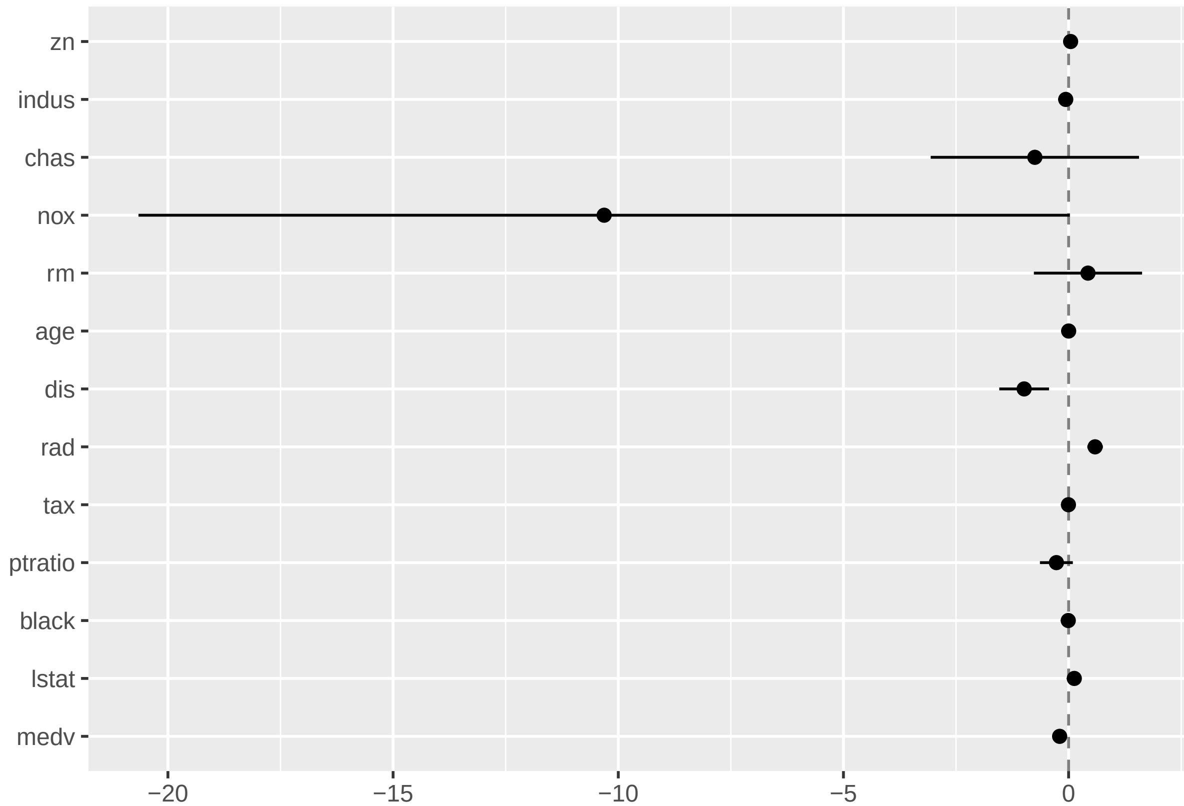
```r
lm_fit <- lm_spec %>%
  fit(crim ~ ., data = Boston)
```

```r
lm_fit %>%
  pluck("fit") %>%
  summary()
```

```
##
## Call:
```

```
## stats::lm(formula = crim ~ ., data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value        Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354        0.018949 *
## zn            0.044855   0.018734   2.394        0.017025 *
## indus        -0.063855   0.083407  -0.766        0.444294
## chas         -0.749134   1.180147  -0.635        0.525867
## nox         -10.313535   5.275536  -1.955        0.051152 .
## rm            0.430131   0.612830   0.702        0.483089
## age           0.001452   0.017925   0.081        0.935488
## dis          -0.987176   0.281817  -3.503        0.000502 ***
## rad           0.588209   0.088049   6.680 0.0000000000646 ***
## tax          -0.003780   0.005156  -0.733        0.463793
## ptratio      -0.271081   0.186450  -1.454        0.146611
## black        -0.007538   0.003673  -2.052        0.040702 *
## lstat         0.126211   0.075725   1.667        0.096208 .
## medv         -0.198887   0.060516  -3.287        0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 0.00000000000000022
```

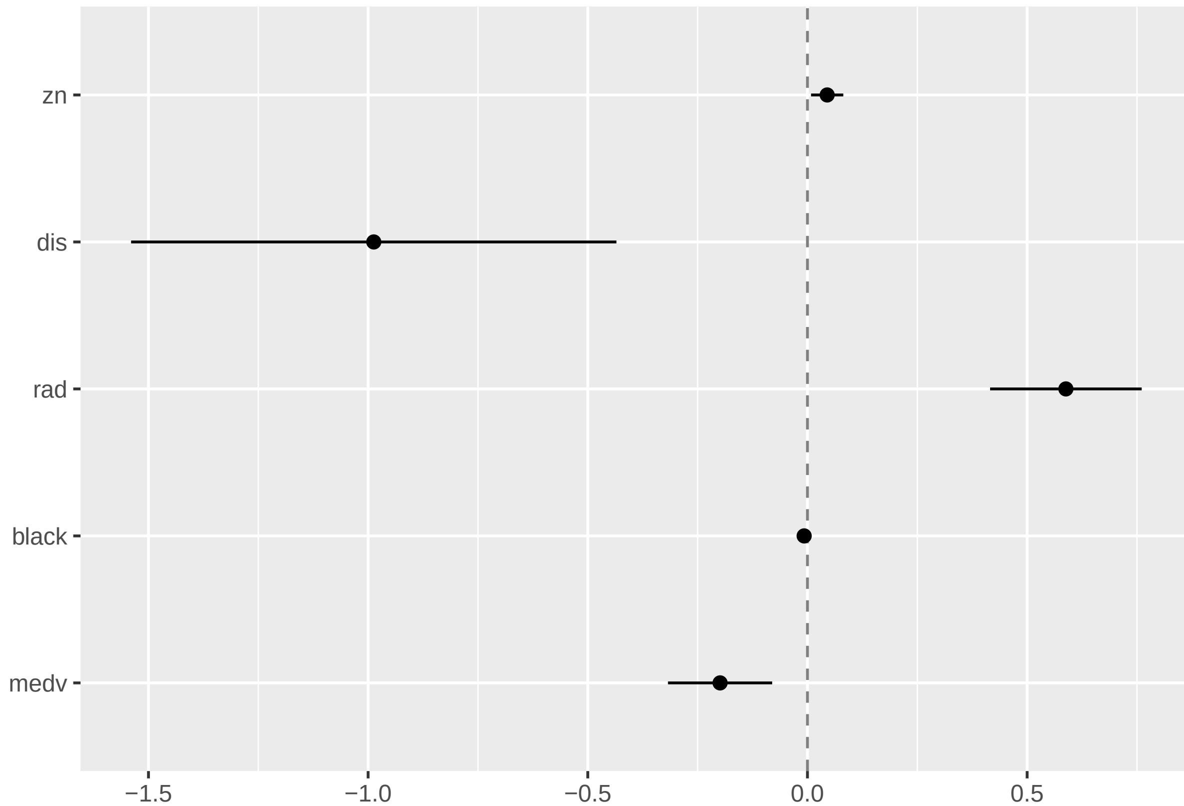
```
tidy(lm_fit) %>%
  dwplot(dot_args = list(size = 2, color = "black"),
         whisker_args = list(color = "black"),
         vline = geom_vline(xintercept = 0, colour = "grey50", linetype = 2))
```




```
significant_predictors <- tidy(lm_fit) %>%
  filter(p.value < 0.05)
significant_predictors
```

```
## # A tibble: 6 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) 17.0       7.23         2.35 1.89e- 2
## 2 zn           0.0449    0.0187       2.39 1.70e- 2
## 3 dis         -0.987     0.282       -3.50 5.02e- 4
## 4 rad          0.588     0.0880       6.68 6.46e-11
## 5 black       -0.00754   0.00367     -2.05 4.07e- 2
## 6 medv        -0.199     0.0605      -3.29 1.09e- 3
```

```
significant_predictors %>%
  dwplot(dot_args = list(size = 2, color = "black"),
         whisker_args = list(color = "black"),
         vline = geom_vline(xintercept = 0, colour = "grey50", linetype = 2))
```




```
??Boston
```

As we can see now only 5 predictors are significant (p value above 0.05).

`zn`: proportion of residential land zoned for lots over 25,000 sq.ft. `dis` weighted mean of distances to five Boston employment centres. `rad` index of accessibility to radial highways. `black` 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town. `medv` median value of owner-occupied homes in $1000s.

Exercise 3

```
#install.packages("wooldridge")
library(wooldridge)
```

```
##
## Attaching package: 'wooldridge'

## The following object is masked from 'package:MASS':
##
##     cement
```

```
data(hprice1)
```

```
??hprice1
head(hprice1)
```

```
##     price assess bdrms lotsize sqrft colonial   lprice  lassess llotsize
## 1 300.000  349.1     4    6126  2438        1 5.703783 5.855359 8.720297
## 2 370.000  351.5     3    9903  2076        1 5.913503 5.862210 9.200593
## 3 191.000  217.7     3    5200  1374        0 5.252274 5.383118 8.556414
## 4 195.000  231.8     3    4600  1448        1 5.273000 5.445875 8.433811
```

```
## 5 373.000  319.1     4    6095  2514         1 5.921578 5.765504 8.715224
## 6 466.275  414.5     5    8566  2754         1 6.144775 6.027073 9.055556
##     lsqrft
## 1 7.798934
## 2 7.638198
## 3 7.225482
## 4 7.277938
## 5 7.829630
## 6 7.920810
```

```
lm_fit <- lm_spec %>%
  fit(price ~ sqrft + bdrms, data = hprice1)
```

```
hprice1_pred <- bind_cols(
    lm_fit %>% predict(new_data = hprice1),
    hprice1
  )
hprice1_pred[1,]
```

```
## # A tibble: 1 x 11
##   .pred price assess bdrms lotsize sqrft colonial lprice lassess llotsize lsqrft
##   <dbl> <dbl>  <dbl> <int>   <dbl> <int>    <int>  <dbl>   <dbl>    <dbl>  <dbl>
## 1  355.   300   349.     4    6126  2438        1   5.70    5.86     8.72   7.80
```

The predicted selling price for the first house is 354.6052.