Homework 1 Solutions

STAT 3032

PROBLEMS: 1.1, 1.6, 2.1(1-3), 2.2, 2.4, 2.13, 2.16(1-4)

CHAPTER 1

1.1

1. The predictor is a function of `ppgdp`, and the response is a function of `fertility`.
2.

```
library("alr4")
```

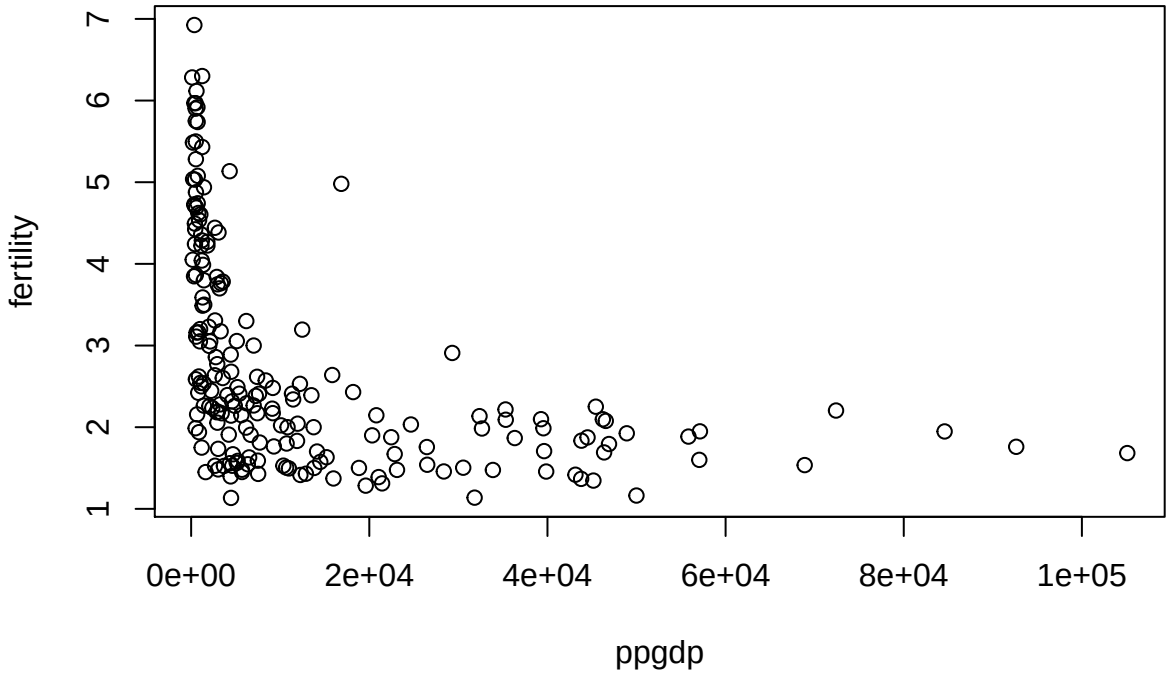
```
## Loading required package: car
```

```
## Loading required package: effects
```

```
##
## Attaching package: 'effects'
```

```
## The following object is masked from 'package:car':
##
##     Prestige
```

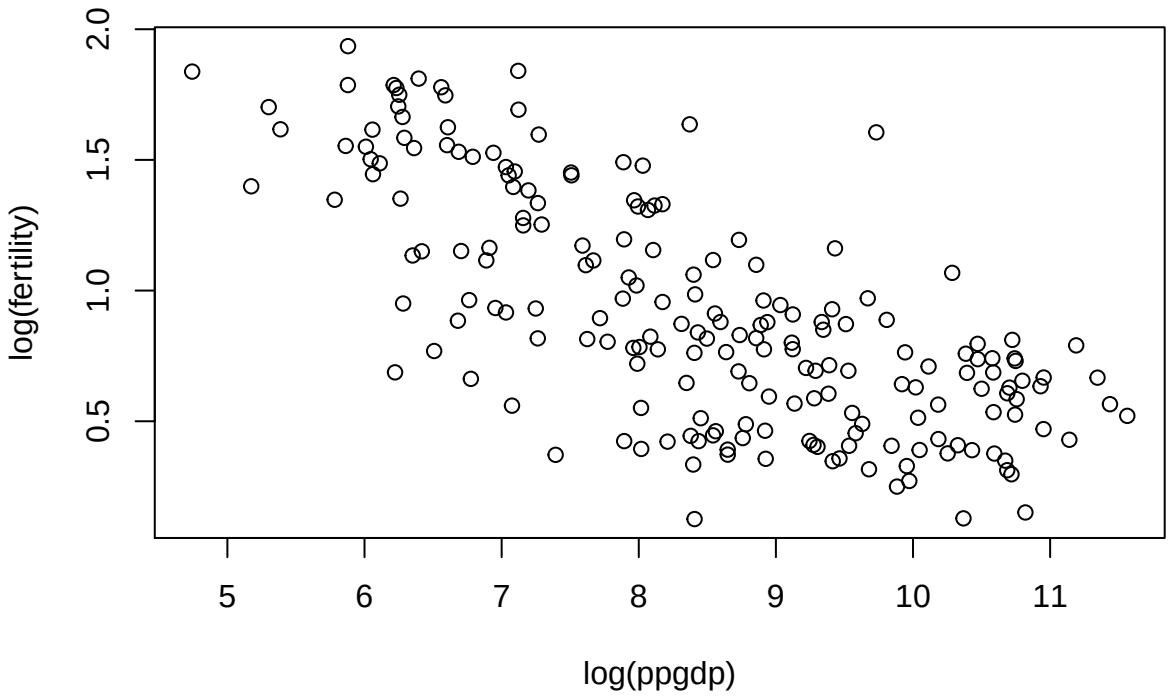
```
data = UN11

plot(fertility ~ ppgdp, data = data)
```



We see that simple linear regression is not a good summary of this graph. the mean function does not appear to be linear, variance does not appear to be constant.

3.

```
plot(log(fertility) ~ log(ppgdp), data = data)
```



Simple linear regression is much more appropriate in log-scale, as the mean function appears to be linear, and constant variance across the plot is at least plausible, if not completely certain. As one might expect, there may be a few outliers that are localities with either unusually high or low `fertility` for their value of `ppgdp`.

1.6

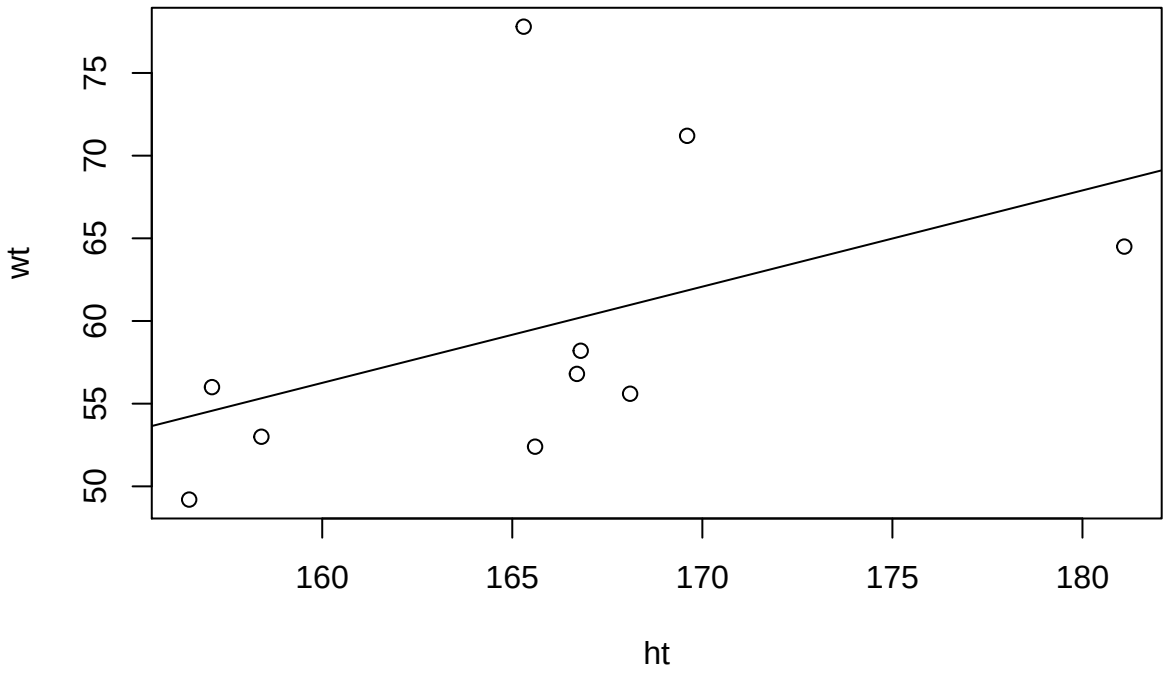
Remarkably, `quality`, `helpfulness`, and `clarity` are almost perfectly related: all three can be taken to measure the same thing. The `easiness` scale is related weakly, but positively related to the first three scales, while `raterIntercept` appears to be mostly unrelated to the other scales.

CHAPTER 2

2.1(1 - 3)

1.

```
data = Htwt
plot(wt ~ ht, Htwt)
abline(lm(wt ~ ht, Htwt))
```



With only 10 points, judging the adequacy of the model is hard, but it may be plausible here, as the value of the response is generally increasing from right to left, and the straight line on the plot is visually a plausible summary of this trend.

2.

```
n = dim(data)[1]
ave = colMeans(data)
ave
```

```
##     ht     wt
## 165.52  59.47
```

```
xbar = ave[1]
ybar = ave[2]
crossprod = (dim(data)[1] - 1)* cov(data)
crossprod
```

```
##         ht      wt
## ht 472.076 274.786
## wt 274.786 731.961
```

```
SXX = crossprod[1,1]
SYY = crossprod[2,2]
SXY = crossprod[1,2]
SXX
```

```
## [1] 472.076
```

```
SYY
```

```
## [1] 731.961
```

```
SXY
```

```
## [1] 274.786
```

The matrix `crossprod` has `SXX` and `SYY` on the diagonal and `SXY` as either off-diagonal entry.

3.

```
#We use the computations from the last subproblem. We do the coefficient estimates first:
coefs = c(Intercept = ybar - (SXY/SXX) * xbar, Slope = SXY/SXX)
coefs
```

```
## Intercept.wt        Slope
##    -36.87588      0.58208
```

```
#Next, we estimate the variance:
s2 = (SYY - SXY^2/SXX)/(n - 2)
s2
```

```
## [1] 71.5017
```

```
#Finally, standard errors of the coefficients and `t-values`:
secoefs = c(Intercept = sqrt(s2 * (1/n + xbar^2/SXX)), Slope = sqrt(s2 * (1/SXX)))
secoefs
```

```
## Intercept.ht        Slope
##   64.4728000    0.3891815
```

```
cov12 = -s2 * xbar/SXX
cov12
```

```
##        ht
## -25.07003
```

```
tvals = coefs/secoefs
tvals
```

```
## Intercept.wt        Slope
##   -0.5719603    1.4956517
```

2.2

1. Points above the line are cities for which prices increased over the time period, and for cities below the line prices decreased.
2. Vilnus and Mumbai, respectively.
3. From (2.5), $\hat{\beta}_1 = rSD_y/SD_z$, and for these data $SD_z \approx SD_y$:

```
data =  UBSprices
apply(data[, c("rice2003", "rice2009")], 2 , sd)
```

```
## rice2003 rice2009
## 14.64513 14.76381
```

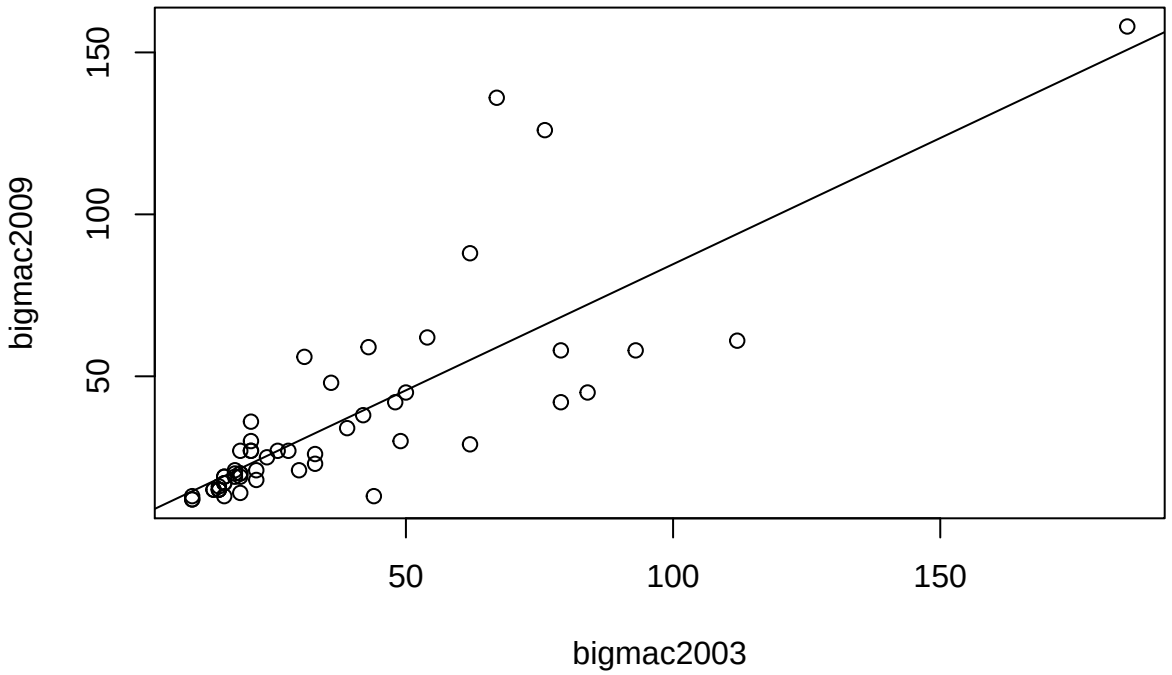
Note that if these were exactly equal, then the slope estimate would equal the correlation and must therefore be between -1 and 1, and couple equal 1 only if prices did not change in any city over the time period.

4. (1) variability is much higher at the right of the graph than at the left; (2) outliers are apparent; (3) of lesser importance, but still suggestive, is that the values on the two axes are clearly skewed, with many small values and a few larger ones.

2.4

1.

```
data = UBSprices
plot(bigmac2009 ~ bigmac2003, data = data)
fitted = lm(bigmac2009 ~ bigmac2003, data = data)
abline(fitted)
```

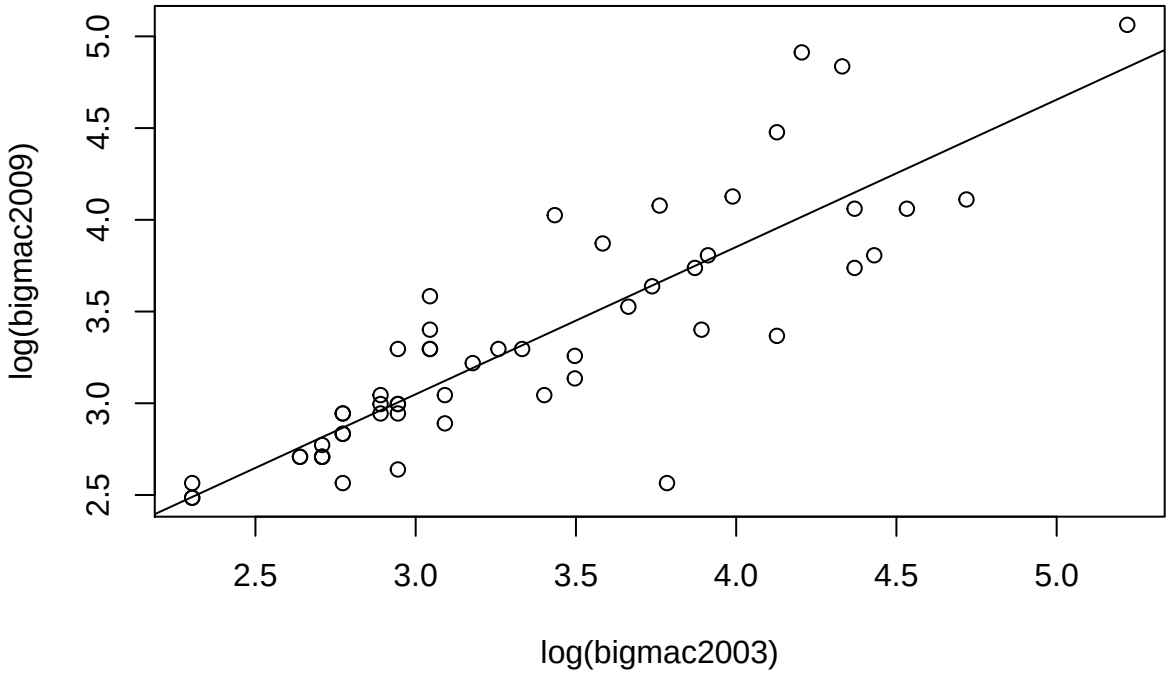


Nairobi was expensive in both years, while `Jakarta` and `Caracas` were relatively expensive in 2009. the price fell relatively far in `Mumbai`.

2. (1) variability is higher at the right of the graph than at the left; (2) outliers were apparent; (3) of lesser important, but still suggestive, is that the values on the two axes are clearly skewed.

3.

```
data = UBSprices
plot(log(bigmac2009) ~ log(bigmac2003), data = data)
fitted = lm(log(bigmac2009) ~ log(bigmac2003), data = data)
abline(fitted)
```



The log-scale graph appears nearly linear, the distribution of the points on the axes are no longer skewed, and variability appears constant. Also the points at the extreme right are no longer separated from the other points. `Jakarta` and `Warsaw` appear to be outliers.

2.13

1.

```
data = Heights
colMeans(Heights)
```

```
##  mheight  dheight
## 62.45280 63.75105
```

```
var(Heights)
```

```
##          mheight  dheight
## mheight 5.546511 3.004806
## dheight 3.004806 6.760274
```

```
m1 = lm(dheight ~ mheight, data = data)
summary(m1)
```

```
##
## Call:
## lm(formula = dheight ~ mheight, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.397 -1.529  0.036  1.492  9.053
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.91744    1.62247   18.44   <2e-16 ***
## mheight      0.54175    0.02596   20.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.266 on 1373 degrees of freedom
## Multiple R-squared:  0.2408, Adjusted R-squared:  0.2402
## F-statistic: 435.5 on 1 and 1373 DF,  p-value: < 2.2e-16
```

The `t-statistic` for the slope has a p-value very close to 0, suggesting strong that $\beta_1 \neq 0$. The value of $R^2 = 0.241$, so only about one-fourth of the variability in daughter's height is explained by mother's height.

2. Although the confidence intervals can be computed from the formulae in the text, most programs will produce them automatically. In R the function `confint` does this:

```
confint(m1, level = 0.99)
```

```
##                  0.5 %     99.5 %
## (Intercept) 25.7324151 34.1024585
## mheight      0.4747836  0.6087104
```

3.

```
predict(m1 , data.frame(mheight = 64), interval = "prediction", level = 0.99)
```

```
##        fit      lwr      upr
## 1 64.58925 58.74045 70.43805
```

2.16(1 - 4)

1.

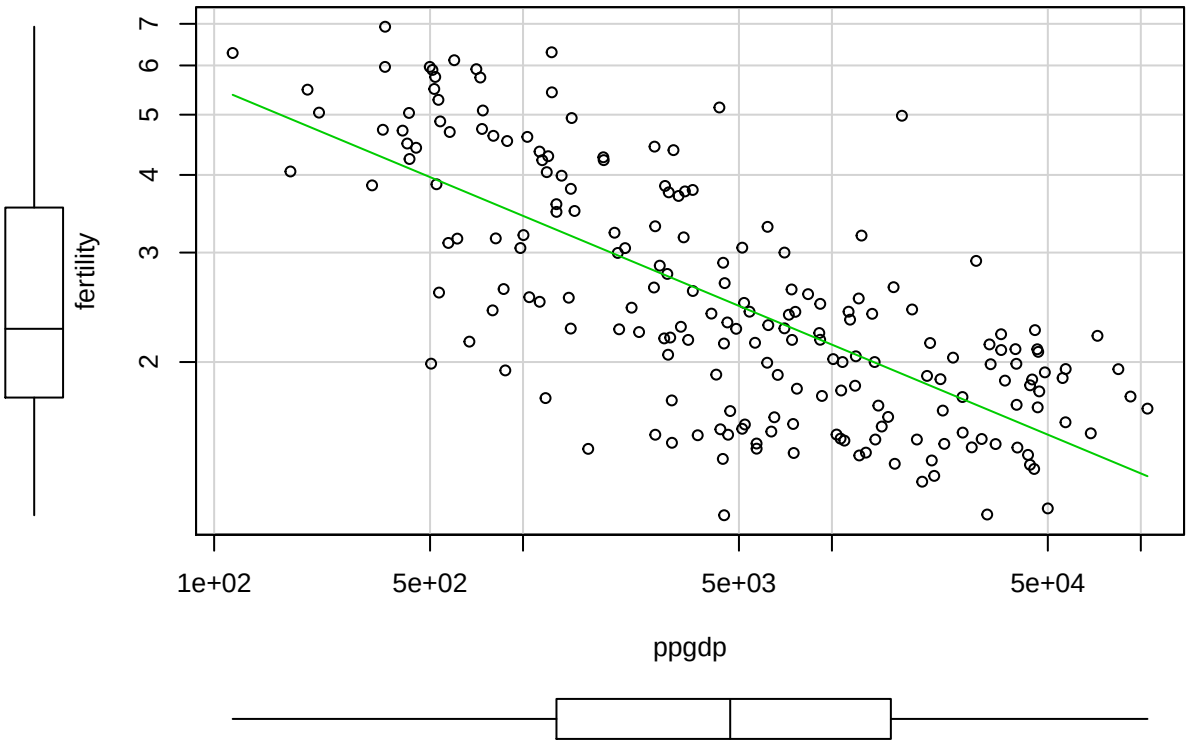
```
data = UN11
m1 = lm(log(fertility) ~ log(ppgdp), data = data)
summary(m1)
```

```
##
## Call:
## lm(formula = log(fertility) ~ log(ppgdp), data = data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79828 -0.21639  0.02669  0.23424  0.95596
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.66551    0.12057   22.11   <2e-16 ***
## log(ppgdp)  -0.20715    0.01401  -14.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3071 on 197 degrees of freedom
## Multiple R-squared:  0.526,  Adjusted R-squared:  0.5236
## F-statistic: 218.6 on 1 and 197 DF,  p-value: < 2.2e-16
```

2.

```
scatterplot(fertility ~ ppgdp, data = data, log = "xy", smooth = FALSE)
```



The `scatterplot` function always draws the fitted line unless you suppress it using the argument `reg.line = FALSE`. You could suppress the boxplots with the argument `boxplots = FALSE`.

Alternatively, you can get the same graph, but with the ticks labeled in log-units, using `scatterplot(log(fertility) ~ log(ppgdp), data = UN11, smooth = FALSE)`.

3. The t-test can be used, $t = -14.79$ with 197 df. the p-value is essentially 0, so the one-sided p-value will also be near 0. We have strong evidence that $\beta_1 < 0$ suggesting that countries with higher `log(ppgdp)` have on average lower `log(fertility)`.

4. \$R\^{2} = 0.526, so about 52.6% of the variability in `log(fertility)` can be explained by conditioning on `log(ppgdp)`.