Problem Set 5 Solution Attempt

Lars L. Ankile

March 31, 2020

Problem 1.

When conducting multiple tests, the α must be adjusted according to the number of individual tests he performs. If you're using $\alpha = 0.05$, and you're doing 200 tests, then you can expect to get false positive results, i.e. reject the null, 10 times. This is not acceptable. Therefore, the α he should operate with could e.g. be $\alpha/200 = 0.00025$ to account for this. Also, he really should report all his results. If you only report your significant results it's very hard to replicate the studies, and it's actually considered pretty bad practice. This is from what I can gather an example of what is called p-hacking.

Problem 2.

a) From using the `power.t.test`-command, I get that there should be $\lceil 304.872 \rceil = 305$ people per group, i.e. $2 \cdot 305 = 610$ people in total.

```
# Calculating the number of participants per group with power.t.test
power.t.test(n = NULL, delta = 0.5, sd = 2.2,
             sig.level = 0.05, power = 0.80)
```

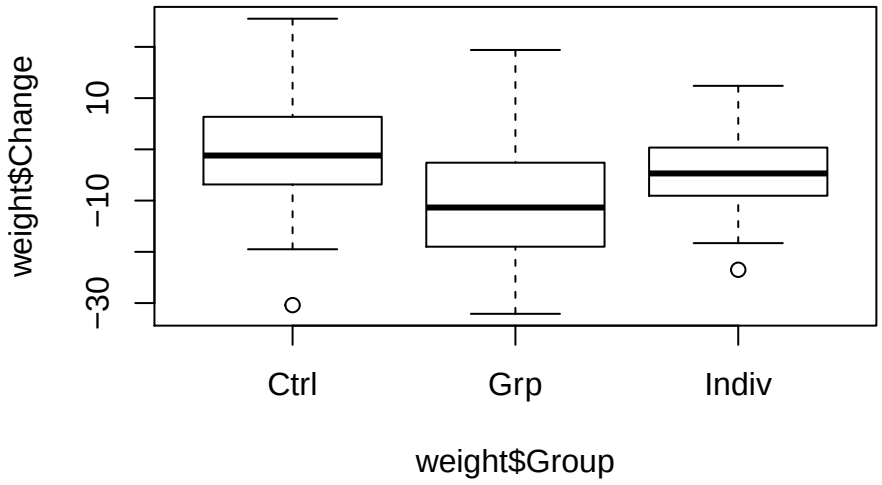
```
##
##      Two-sample t test power calculation
##
##               n = 304.872
##           delta = 0.5
##              sd = 2.2
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

b) By increasing α to say 0.01, one would need a lot more study participants to achieve the same power. That is because we require a much higher level of evidence for rejecting the null which means that even though the alternative is true we won't necessarily get strong enough evidence for it with smaller α which lessens the power.

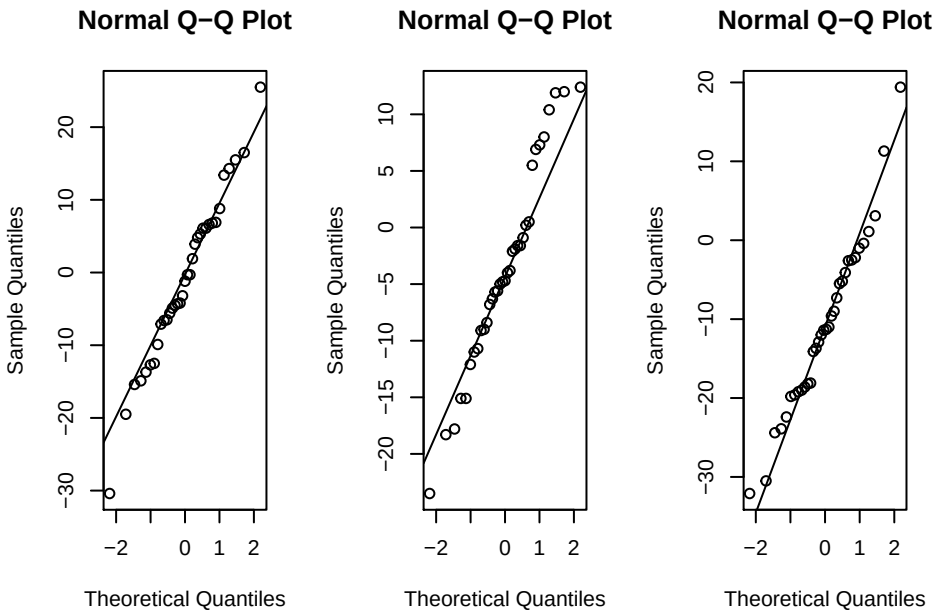
Problem 3.

a) From the below boxplot, we can observe that there seems to be relatively large differences between the different groups. The control group had almost no median change in weight, while the individual incentive group has a median around 5 pounds lost, and the group incentive group around 10. The control and group incentive group both have simialar length of whiskers, while the individual incentive group has much shorter whiskers, i.e. there's less variability in that group for some reason.

```
# Load the data
weight <- read.csv("datasets/weight.csv")

# Create plot
boxplot(weight$Change ~ weight$Group)
```




```
par(mfrow = c(1, 3))
qqnorm(weight$Change[weight$Group == 'Ctrl'])
qqline(weight$Change[weight$Group == 'Ctrl'])
qqnorm(weight$Change[weight$Group == 'Indiv'])
qqline(weight$Change[weight$Group == 'Indiv'])
qqnorm(weight$Change[weight$Group == 'Grp'])
qqline(weight$Change[weight$Group == 'Grp'])
```



b) i. There's three assumptions that should be met. (1) Observations should be independent across the different groups, (2) Observations within each group should be roughly normal, and (3) variability between the groups should be about equal. The first assumption is most likely met since the groups are randomized. The second assumption seems more or less satisfied from inspecting both the boxplots and the qqplots. It's not perfect, but likely close enough. Assumption 3 seems also satisfied by inspecting the boxplots, but should be verified numerically. The largest variance is 132.3, which is less than three times the smallest, which is 82.4, so we're in good shape assumption-wise.

ii. From the below analysis we see that the probability that all the means are the same is less than 0.001, i.e. there's strong evidence against the null hypothesis. From conducting a pairwise t-test, and using significance level $\alpha = 0.05$, we see that there are significant differences between both the control and the group incentive group and the individual incentive group and the group incentive group. The adjustted p-value is just below the threshold for the latter groups, and extremely low for the former groups. From this we can conclude with relatively high confidence that economic incentives work and especially in a group setting.

```
# Check assumptions
# Check the variances
var(weight$Change[weight$Group == 'Ctrl'])
```

```
## [1] 132.2667
```

```
var(weight$Change[weight$Group == 'Grp'])
```

```
## [1] 124.0807
```

```
var(weight$Change[weight$Group == 'Indiv'])
```

```
## [1] 82.41669
```

```r
# Do ANOVA analysis
summary(aov(weight$Change ~ weight$Group))
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## weight$Group   2   1753   876.3   7.768 0.000728 ***
## Residuals    101  11394   112.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Do a pairwaise t-test
pairwise.t.test(weight$Change,
                weight$Group,
                p.adjust.method = "bonf")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  weight$Change and weight$Group
##
##       Ctrl    Grp
## Grp   0.00069 -
## Indiv 0.87037 0.02019
##
## P value adjustment method: bonferroni
```

c) i. I think this would cause a much larger spread in the data for the group incentive group. That is because I think some people would be very motivated by that and would work very hard to make sure they win each week. Others's, I suspect, would automatically give upo because they know woth themselves that they would never be able to compete for the price anyways. So the competitive people within group would probably lose more weight, while the less competitive people would lose less weight.

ii. I don't think it would advisable, because the goal should be to help as many people lose weight as possible. This way incentivices a few people to lose a lot, while the large majority probably won't lose as much. Also, this kind of competition is probably not as healthy and sustainable over the long-term. Personally, I have more belief in the sense of community and the support that can create for the participants with other models.

Problem 4.

a) i. The conclusion of the below simulation and t-test is that the difference in means between the groups is not signifcant with $p = 0.4243$, i.e. much larger than the significance level if using $\alpha = 0.05$ for example.

```
# Set parameters
mean_control <- 140
mean_treat <- 138
sd_both <- 10
n <- 25

# Set the seed
set.seed(666)

# Simulate values
obs_control <- rnorm(n, mean = mean_control, sd = sd_both)
obs_treat <- rnorm(n, mean = mean_treat, sd = sd_both)

# Conduct test
t.test(obs_control, obs_treat)
```

```
##
##  Welch Two Sample t-test
##
## data:  obs_control and obs_treat
## t = 0.80599, df = 46.646, p-value = 0.4243
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.903510  9.120414
## sample estimates:
## mean of x mean of y
##  139.6562  137.0477
```

ii. The conclusion in part i. is indeed a Type II error becuase we failed to reject the null even though the alternative hypothesis is true.

b) i. From running the below simulation with 1000 replicates, I get that the power of the study with 25 participants in each study group is 11.2%. That is very weak power and means that we will expect to get true positives 11.2% of the time and not reject the false null 88.8% of the time.

```
# Create a function for less code duplication and easier reuse later
estimate_power <- function(n, sd_both = 10, mean_treat = 138) {

  # Set parameters
  alpha <- 0.05
  mean_control <- 140
  replicates <- 1000
```

```
  # Set the seed
  set.seed(666)

  #create empty vectors
  results <- vector(mode = "logical", lengt = replicates)

  #run repeated tests and record p-values

  for (k in 1:replicates) {

    # Simulate values
    obs_control <- rnorm(n, mean = mean_control, sd = sd_both)
    obs_treat <- rnorm(n, mean = mean_treat, sd = sd_both)

    # Conduct test
    results[k] <- t.test(obs_control, obs_treat)$p.value < alpha
  }

  # Estimate power
  return (sum(results) / length(results))
}
```

```
estimate_power(n = 25)
```

```
## [1] 0.112
```

ii. Below I've done the same simulation as above three additional times, with size of each study group increased to 50, 100, and 200, respectively. We see that this increases the power to 16.5%, 32.3%, and 52.5%, respectively. Still not good, but a lot better than before.

```
estimate_power(n = 50)
```

```
## [1] 0.168
```

```
estimate_power(n = 100)
```

```
## [1] 0.323
```

```
estimate_power(n = 200)
```

```
## [1] 0.525
```

c) i. I'd expect the power to decrease if there's more variations in the observations. This is because if you have more variation, the probability that the observed mean takes on more extreme values by chance increases. Another way to see it is that the distributions for the null hypothesis and the alternative hypothesis will have larger overlap if the distributions are more spread out, and so it's harder to reject the null.

ii. I've reused the function from b), just with different parameters to run this simluation. We see that when we run the simulation for 100 participants in each group with standard deviation of 5, 10, and 15, we get an estimated power of 81.2%, 32.3%, and 15.5%, respectively. These are large differences. The group with the lowest standard deviation

has an estimated power above 80%, whixh is starting to be pretty strong. The other two are very weak.

```
# Simulate the estimated power with different standard deviations
# Number of study participants is set to 100
estimate_power(n = 100, sd_both = 5)
```

```
## [1] 0.812
```

```
estimate_power(n = 100, sd_both = 10)
```

```
## [1] 0.323
```

```
estimate_power(n = 100, sd_both = 15)
```

```
## [1] 0.155
```

d) i. If the true difference in means is relatively large I'd expect the probability of rejecting the mean to be relatively higher than if the true difference was small.

ii. Again, I've used the function from b) to run the simulation. I've also used $n = 100$ here for all the simulations because I think the results are more interesting for that study group size. For a true mean of 138, 137, and 136 mmHg for the treatment group, we see an estimated power of 32.3%, 57.4%, and 81.2% respectively. Here we see that the power increases pretty drastically with a slight increase in effect size, and with effect size of 2 mmHg we get power above 80% for 100 participants per group with a standard deviation of 10 mmHg.

```
estimate_power(n = 100, mean_treat = 138)
```

```
## [1] 0.323
```

```
estimate_power(n = 100, mean_treat = 137)
```

```
## [1] 0.574
```

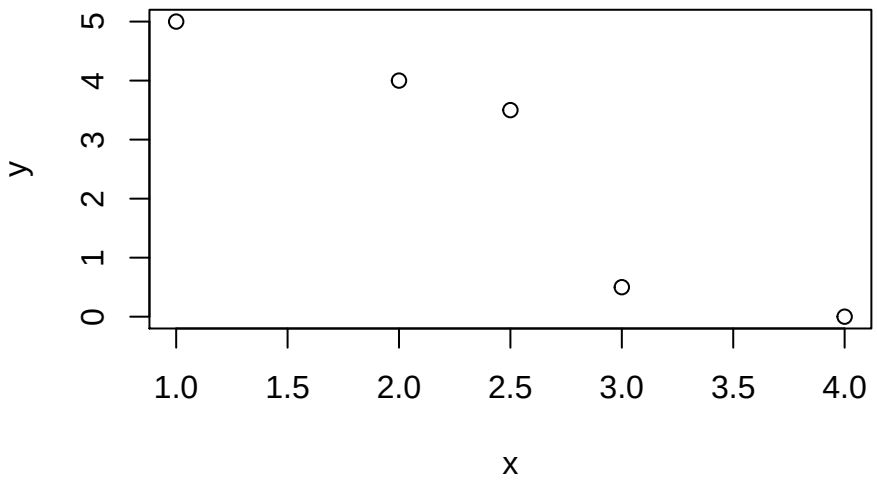
```
estimate_power(n = 100, mean_treat = 136)
```

```
## [1] 0.812
```

Problem 5.

a) Below you can see a scatter plot of the data.

```
# Define the data
x <- c(1, 2, 2.5, 3, 4)
y <- c(5, 4, 3.5, 0.5, 0)

# Plot the data
plot(y ~ x)
```



b) Below I've calculated the correlation between x and y by using R as an calculator to calculate the standard formulas for correlation. The correlation value I got was -0.932, which means there's a pretty strong negative correlation, i.e. when x increases, y decreases, and the data lies pretty close to a straight line.

```
# Use r as a calculator
x_bar <- sum(x) / length(x)
y_bar <- sum(y) / length(y)

s_x <- sqrt(sum((x - x_bar)^2) / (length(x) - 1))
s_y <- sqrt(sum((y - y_bar)^2) / (length(y) - 1))

r <- 1 / (length(x) - 1) * sum((x - x_bar) * (y - y_bar)) / (s_x * s_y)
r
```

```
## [1] -0.9320165
```

c) Below I've used R as an calculator to calculate the slope and intercept of the regression line. The slope came out to be $b_1 = -1.85$ and the intercept came out to be $b_0 = 7.225$.

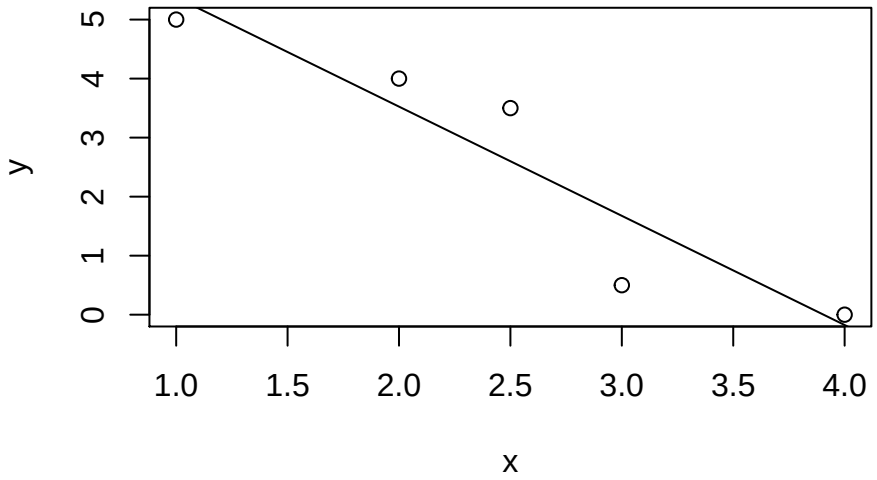
```
# Use r as a calculator
b1 <- r * s_y / s_x
b0 <- y_bar - b1 * x_bar

b1; b0
```

```
## [1] -1.85
```

```
## [1] 7.225
```

d) Below, I've drawn in a straight line with the slope and intercept found above, together with the points plotted in a). We can see that the points all are relatively close to the line, but not perfectly on it.

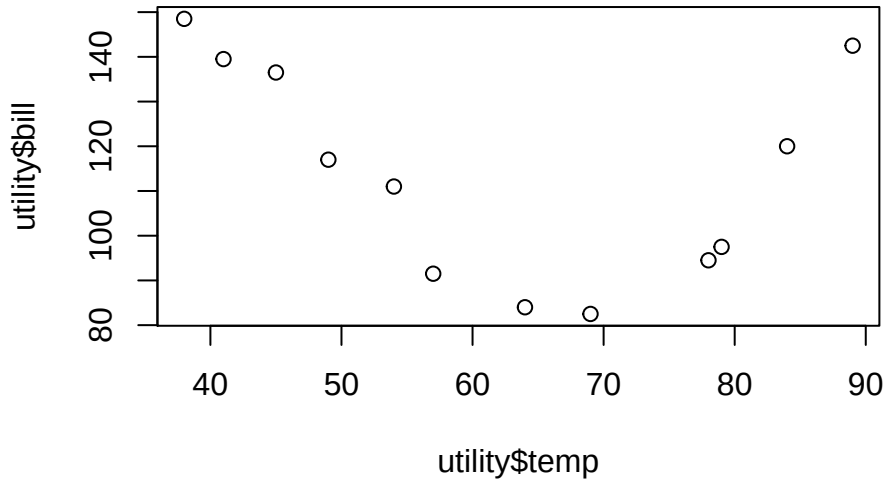
```
# Plot the data with the regression line
plot(y ~ x)
abline(a = b0, b = b1)
```


y
0
1
2
3
4
5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x

Problem 6.

a) Below is a scatter plot with the average temperature for a month on the x-axis and the size of the utility bill in dollars for the same month on the y-axis. There doesn't seem to be a linear relationship between the variables, because the bills decrease in size when you go from average temperature of 35 to 65, but it increases from 65 to 90. However, first and second part each on their own seems to be a pretty linear relationship, and one could probably model this with two distinct regression lines, and just use the appropriate one according to what temperature it is at any given time.

```
# Load the data
utility = read.csv("http://people.fas.harvard.edu/~mparzen/stat104/utility.csv")

# Create a plot
plot(utility$bill ~ utility$temp)
```



b) Below I've found the slope and the intercept of the fitted model by using the `lm`-command. By using these values and the temperature of 120 degress, I get that the predicted utility bill would be \$86.93. This makes little sense, seeing that once you cross degrees the bill exceeds \$140, and it had been increasing rapidly up t that point too. You can also see the below scatter plot with the fitted line also in the plot. The line is slowly sloping downward over the whole interval, and is clearly a bad fit for the points beyond 85 degrees.

```
# Fit a model
coef <- lm(utility$bill ~ utility$temp)$coefficients
coef
```

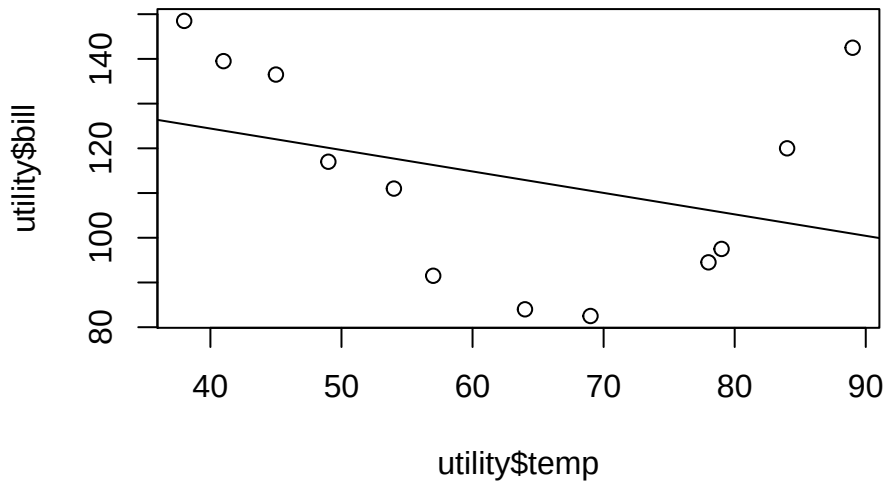
```
##  (Intercept) utility$temp
##  143.6228051   -0.4798844
```

```
b1 <- coef['utility$temp']
b0 <- coef['(Intercept)']

# Calculate estimate
temp <- 120
```

```r
as.numeric(b0 + temp * b1)
```

```
## [1] 86.03667
```

```r
plot(utility$bill ~ utility$temp)
abline(a = b0, b = b1)
```



Problem 7.

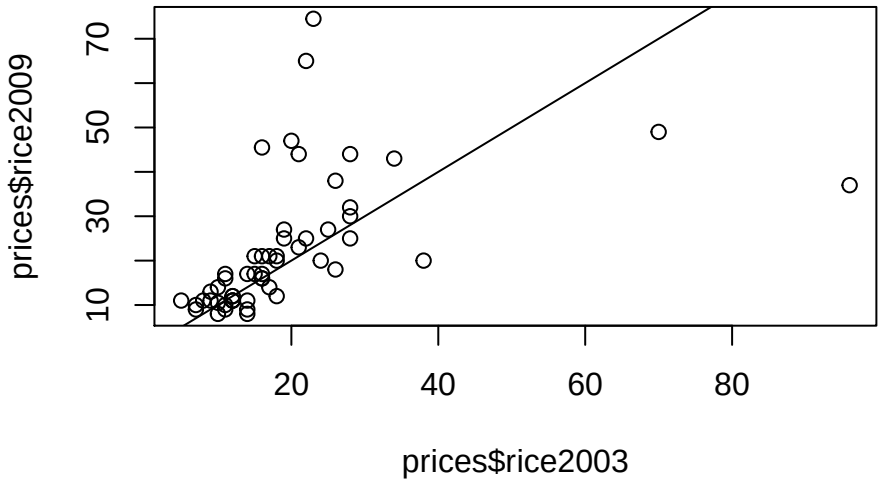
a) The result of such a regression will give $b_0 = 40$, $b_1 = -1$, and $R^2 = 0$. This is because if a student have no right answers, they must have exactly 40 wrong answers. One right answer means 39 wrong answers, 2 right, 38 wrong, and so on. This follows a perfect, straight line from 40 on the y-axis to 40 on the x-axis.

b) In my opinion, this is not a useful model to fit, because we already know the underlying process creating the observed data. In such cases, it's bettery to just use the known relationship between the variables directly instead of trying to fit a line to it.

Problem 8.

a)

```
# Load the data
prices = read.csv("datasets/ubsprices.csv")

# Plot the data and the y = x line
plot(prices$rice2009 ~ prices$rice2003)
abline(a = 0, b = 1)
```



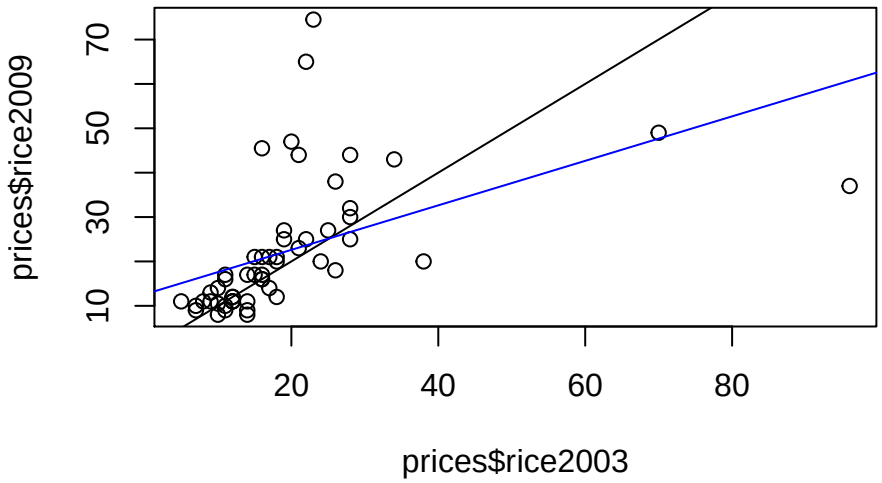
b) For points above the $y = x$-line, the price of rice has increased in the period 2003 to 2009. For points below the line the prices of rice have decreased from 2003 to 2009.

c) The slope coefficient in the fitted linear blue line below is $b_1 = 0.501$, i.e. about half of the diagonal $y = x$-line. This means that the tendency for all countries overall is for the price of rice to fall from 2003 to 2009, except for a few cases to the left of about $x = 24$, where the prices are predicted to rise from 2003 to 2009.

```
# Fit a linear model
coef <- lm(prices$rice2009 ~ prices$rice2003)$coef
coef
```

```
##     (Intercept) prices$rice2003
##      12.5841920       0.5013831
```

```
plot(prices$rice2009 ~ prices$rice2003)
abline(a = 0, b = 1)
abline(a = coef['(Intercept)'], b = coef['prices$rice2003'], col = 'blue')
```



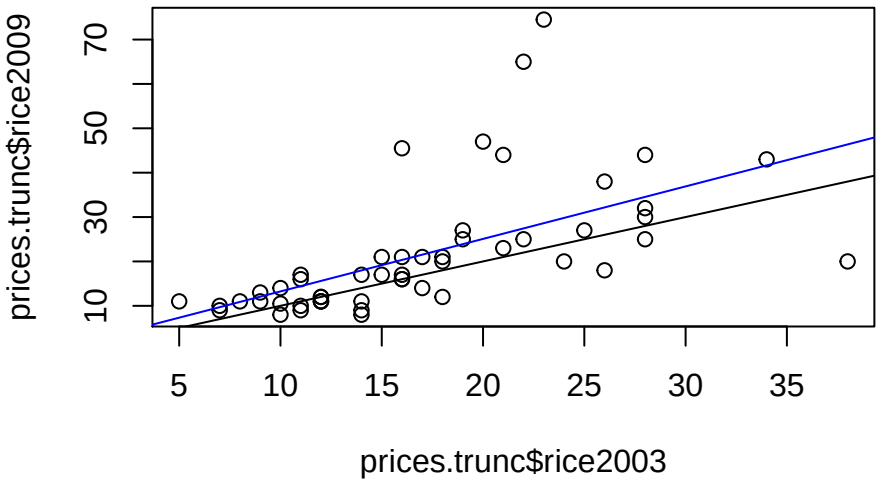
d) There are two points very far to the right in the plot which seems to have had a large impact on the fitted line. Since we're using least-squares to minimize to find the best line, points that are a long way from the line will have a unproportionally larger impact on the value of R^2 than points that are closer, because the distance is squared. These two dots to the right of 60 on the x-axis are both a long way from the $y = x$-line, and we also see that the new fitted line (in blue) is pulled a lot towards those two points.

e) In this new model that excludes the extreme points from above, we suddenly see that the slope has become larger than one, i.e. the model predicts that prices tend to rise over time. Also, the blue fitted line is much closer to the $y = x$-line, just slightly above and rising a little faster. The points therefore seem to have been very influential.

```
# Remove the outliers
prices.trunc <- prices[prices$rice2003 < 60, ]

# Fit new linear model
coef <- lm(prices.trunc$rice2009 ~ prices.trunc$rice2003)$coef
coef
```

```
##           (Intercept) prices.trunc$rice2003
##              1.411498              1.183166
```

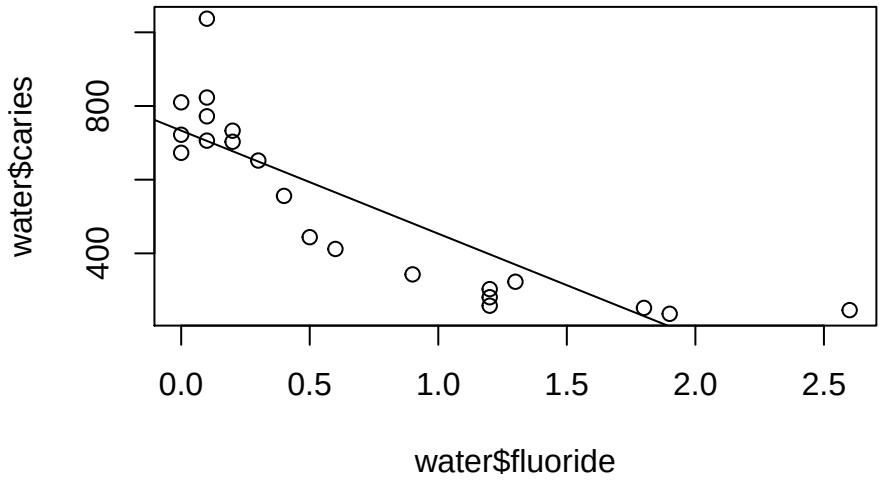
```
# Plot the new model and points
plot(prices.trunc$rice2009 ~ prices.trunc$rice2003)
abline(a = 0, b = 1)
abline(a = coef['(Intercept)'], b = coef['prices.trunc$rice2003'], col = 'blue')
```


prices.trunc$rice2009
10
30
50
70
5
10
15
20
25
30
35
prices.trunc$rice2003

Problem 9.

a) Below is a scatter plot show the relationship between the amount of dental caries in children on the y-axis and the amount of flouride in the water for that child on the x-axis. The linear, fitted line is plotted on top.

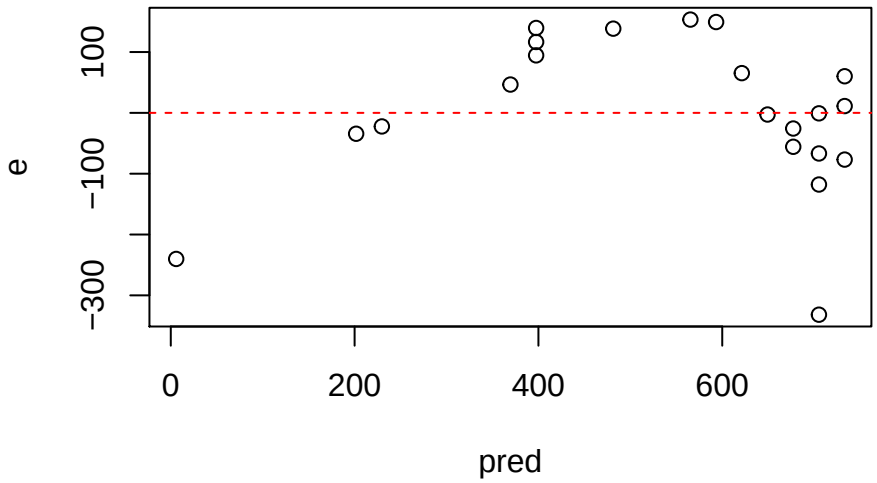
```
# Load the data
load("datasets/water.Rdata")

# Fit a linear model to the data
coef <- lm(water$caries ~ water$fluoride)$coef
b0 <- coef['(Intercept)']
b1 <- coef['water$fluoride']

# Create a plot with the least squares line
plot(water$caries ~ water$fluoride)
abline(a = b0, b = b1)
```



b) By and large, the linear model seems to approximate the points rather well. Still, there are some irregularities, with there being a for example way above the line at roughly 0.2 on the x-axis, and also a point to the right of 2.5. Apart from those, the model seems to fit well, though. It could be that a decaying exponential function would be the best fit in this case.

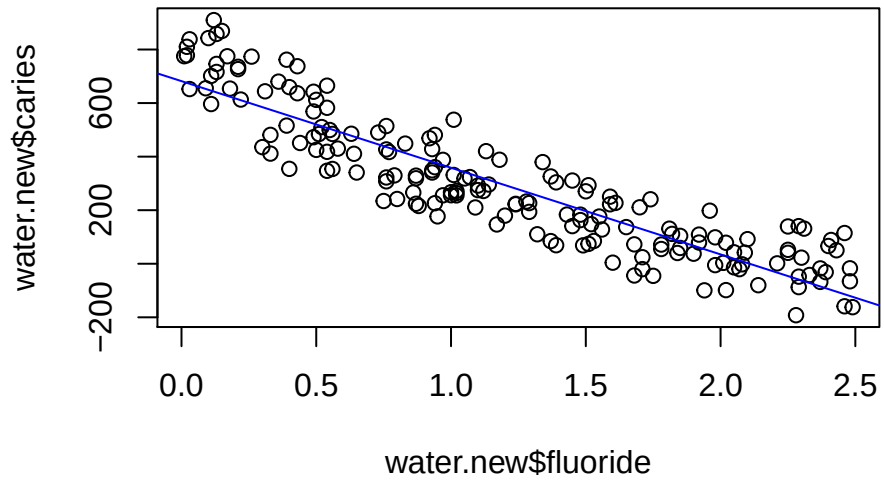
c) The residual plot makes it clearer that the model tends to overshoot values that are in the range 400-600, and undershoot values from 600 and up. This can be seen in the above plot as well, but is more apparent in the residual plot. We see that the assimption of constant variability seems to be violated in this case.

```
# Find predicted values for all x-values based on the linear model
pred <- b0 + b1*water$fluoride
# Calculate all the residual values (differences between predicted and actual)
e <- pred - water$caries

# Residual plot
plot(e ~ pred)
abline(a = 0, b = 0, lty = 2, col = "red")
```



d)

```
# Load the data
load("datasets/water_new.Rdata")

# Fit a linear model to the new water data
coef <- lm(water.new$caries ~ water.new$fluoride)$coef
b0 <- coef['(Intercept)']
b1 <- coef['water.new$fluoride']

# Create a plot with the least squares line
plot(water.new$caries ~ water.new$fluoride)
abline(a = b0, b = b1, col = "blue")
```



e) In this case the data seems to fit very well to the fitted model. It seems like the model lies in the middle of a trend and that there's variability both above and below the line in the whole interval.

f) The below residual plot shows what we saw above in greater detail. The variability does indeed seem to be more constant in this case, and balanced above and below the 0-line. It's slightly prone to overshooting in the midrange of the interval, and slightly prone to undershooting at

the ends of the intval.

```r
# Find predicted values for all x-values based on the linear model
pred <- b0 + b1*water.new$fluoride
# Calculate all the residual values (differences between predicted and actual)
e <- pred - water.new$caries

plot(e ~ pred)
abline(a = 0, b = 0, lty = 2, col = "red")
```