University of Toronto Scarborough
Department of Computer and Mathematical Sciences
STAC32 (K. Butler), Final Exam
December 16, 2023

Aids allowed (on paper, no computers):

- My lecture overheads (slides)
- Any notes that you have taken in this course
- Your marked assignments
- My assignment solutions
- Non-programmable, non-communicating calculator

This exam has 10 numbered pages of questions plus this cover page.
In addition, you have an additional booklet of Figures to refer to during the exam.
The maximum marks available for each part of each question are shown next to the question part.
If you need more space, use the last page of the exam. Anything written on the back of the page will not be graded.
You may assume throughout this exam that the code shown in Figure 1 of the booklet of Figures has already been run.

1. The state of Illinois recorded traffic fatality data for the years 1962 to 1971. The interest is in a possible reduction in deaths when new safety regulations went into effect after 1966. The data file is shown in Figure 2, which is saved in the file `il.txt` in the folder in which R Studio is currently running. The three columns are:

   - `when`: either `before` or `after`, whether before or after the new safety regulations went into effect
   - `year`: the actual year for which the number of deaths was recorded
   - `deaths`: the number of traffic-related deaths in that year, per 100 million vehicle miles.

   Note that when I ask for code in this exam, I want the code and *not* the output; in any case, you usually won't know what the output is going to be.

   (a) [3] What R code would read the data from the file into a dataframe called `illinois` and display that dataframe?

   (b) [2] A graph is shown in Figure 3; what code would produce that graph?

   (c) [2] Look again at the graph is shown in Figure 3. What do you conclude from it that will interest the people that collected the data?

   (d) [2] Explain briefly why the graph in Figure 3 *does not* help us to decide whether to run a t-test or some other test, to see whether the number of deaths has decreased after 1966.

   (e) [2] Some output is shown in Figure 4. What do you conclude, in the context of the data? Explain briefly. (You may assume that it is reasonable to run this test.)

2. Figure 5 shows some code to estimate the power of a test, along with the output from the code. Based on this Figure, answer the following questions.

   (a) [2] What null hypothesis is being tested? Here and elsewhere in this question, if you use any symbols, you should define what they mean.

   (b) [2] What is the alternative hypothesis of the test?

   (c) [3] What population is being sampled from, as far as you can tell from the Figure?

   (d) [1] What size of sample is being taken?

   (e) [3] Explain specifically what the output *means*, for the benefit of a manager who does not know what the code does. (The manager does not want to know what the code *does*; they only care about what the output tells them.)

(f) [2] If I instead used a sample size of 60 (as opposed to the value used in the code in Figure 5), what can you say about what the power would be? Explain briefly.

(g) [3] There is a way to *calculate* the power in this situation, rather than estimating it by simulation. Why is that, and what code would do the calculation?

3. A scientist obtained 24 determinations of the amount of copper in whole wheat flour, in parts per million. It is desired to assess whether the "average" (mean or median, as appropriate) amount of copper is 4 parts per million, or whether it is different from that.

(a) [2] A normal quantile plot of the data is shown in Figure 6. The scientist decided to run a sign test rather than a t-test. How does this Figure, along with anything else you have learned so far about the data, support the scientist's decision? Explain briefly.

(b) [2] What code would run a suitable sign test here? The data values are in a column called `copper` in a dataframe called `flour`.

(c) [3] The output for the sign test for which you gave code in the previous part is shown in Figure 7. What do you conclude, in the context of the data?

(d) [2] How could you have guessed, without looking at the P-values in Figure 7, whether the P-value you wanted would be small or not? Explain briefly.

(e) [2] What code would obtain a confidence interval for the population median copper content?

(f) [3] The bootstrap sampling distribution of the sample mean is shown in Figure 8. How does this support the scientist's decision to use a sign test, and why did this distribution come out with the shape it did?

4. 39 chickens were randomly allocated to one of three different "rations" (feeds), so that 13 chickens received each ration. The weight gain (in kg) of each chicken over a certain time period was recorded. We are interested in whether the different rations are associated with different average (mean or median) weight gains, and, if so, which rations cause the largest weight gains. Some of the data is shown in Figure 9, and a boxplot of the data is shown in Figure 10.

   (a) [3] Three possible analyses are shown in Figures 11, 12, and 13. Which of these analyses do you think is the most suitable for these data? Explain briefly.

   (b) [2] What do you conclude from your chosen analysis, as far as your chosen Figure allows you to conclude?

   (c) [3] Some followup analyses are shown in Figures 14, 15, and 16. Which one is the most appropriate followup for the analysis you used in the previous part, and what do you conclude from it, in the context of the data? If none of those followup analyses are appropriate, explain briefly why.

   (d) [2] Based on what you have found so far, what is your recommendation for the ration or rations to recommend for future use in feeding chickens, given what we are interested in? Justify your choice or choices briefly.

5. This question is about tidying data.

   (a) [2] Dataframe `d1` in Figure 17 contains some data from a two-sample experiment where six individuals were randomly assigned to one of two treatments A and B. What code would transform dataframe `d1` into the dataframe `d2`, shown in Figure 18?

   (b) [2] What statistical purpose might you have for wanting to transform dataframe `d1` into `d2`, as in the previous part?

   (c) [2] What code would transform dataframe `d2`, shown in Figure 18, into dataframe `d1`, shown in Figure 17?

   (d) [3] The data in `d3` (shown in Figure 19) are from an experiment on millipedes. The concentration of a certain amino acid was measured for males and females of each of two species, labelled S1 and S2. Tidy the data to have one column of amino acid values, called `amino`, with other columns labelling the sex and species of each millipede. Do this in the most efficient way.

(e) [3] We want to rearrange the data in `d3` (Figure 19) to have separate columns for the amino acid measurements for males and females (that is, we want columns *named* `male` and `female`, but a column called `species` that *contains* the values `S1` and `S2`). What code will carry out this rearrangement, as directly as possible?

(f) [3] A dataframe `d4` is shown in Figure 20, with some code below it. What output will the code produce?

6. In a study of cheddar cheese from the LaTrobe Valley of Victoria, Australia, samples of cheese were analyzed for their chemical composition and were subjected to taste tests. Overall taste scores were obtained by combining the scores from several tasters. Our aim is to see whether taste score (`taste`) is influenced by any of the three explanatory variables, which are as shown below, measured in suitable units:

   - `Acetic`: concentration of acetic acid
   - `H2S`: concentration of hydrogen sulfide
   - `Lactic`: concentration of lactic acid

   Some of the data are shown in Figure 21.

   (a) [3] Some scatterplots are shown in Figure 22. Describe what these plots tell you.

   (b) [2] In the code at the top of Figure 22, why did I need to use `pivot_longer`? Explain briefly.

(c) [2] A regression model is shown in Figure 23. What regression model would you fit next, using $\alpha = 0.01$? Explain briefly. (If you would not fit any other regression models, explain briefly why not.)

(d) [3] Figures 22 and 23 appear to say something contradictory (that is, what you learn from the two Figures is different in some way). What is it that is contradictory, and what would be a good reason why it happened? Explain briefly.

(e) [3] Some residual plots are shown in Figures 24, 25, and 26. Describe any problems you see in these plots, or say that there are none, as appropriate. In either case, explain how you came to your conclusion.

(f) [2] Some further analysis is shown in Figure 27. What do you conclude from this Figure? Explain briefly.

7. In this course, we had 8 assignments, of which the best 6 are counted towards your grade. There is no function in R to calculate the mean of the largest 6 values out of 8 in a column, such as the column `x` in the dataframe `d` shown in Figure 28. In this question, we will write such a function. (Assume that the assignments shown in Figure 28 have all been graded out of 10 points.)

   (a) [3] What code would make a dataframe that contains the largest 6 out of these 8 scores?

   (b) [3] What code would you add to your code of the previous part to find the mean of the largest six values of `x` *as a number*, and not as a dataframe?

   (c) [3] Write a function called `best` that accepts as input a dataframe containing a column `x`, and outputs the mean of the six largest values in `x` as a number.

(d) [2] What code would run your function with the dataframe shown in Figure 28 as input?

(e) [2] The result of running your function on the data in Figure 28 is shown in Figure 29. (The `[1]` next to the output indicates that the function returns a single value.) By looking at Figure 28, but *without doing any calculation*, how do you know that this result is at least close to being correct?

(f) [2] What code would you add to your function to give an error if the input dataframe was based on something other than eight assignments?

(g) [3] The course grade will usually be based on the best six assignments, but will sometimes be based on a different number. What changes would you make to your function to allow the calculation to be based on a different number of best assignments, but to use the best 6 assignments if the number of best ones to use is not specified? Adapt your code from part (c) for this.

Use this page if you need more space. Be sure to label any answers here with the question and part they belong to.

Numbered Figures begin here:

```
library(MASS)
library(tidyverse)
library(smmr)
library(broom)
```

Figure 1: Packages that are loaded in this exam

when	year	deaths
before	1962	4.9
before	1963	5.1
before	1964	5.2
before	1965	5.1
before	1966	5.3
after	1967	5.1
after	1968	4.9
after	1969	4.7
after	1970	4.2
after	1971	4.2

Figure 2: Illinois traffic deaths data

```
-- Column specification --------------------------------------------------------
cols(
  when = col_character(),
  year = col_double(),
  deaths = col_double()
)
```

Figure 3: Illinois traffic deaths: graph

```
    Welch Two Sample t-test

data:  deaths by when
t = -2.5717, df = 5.0359, p-value = 0.0248
alternative hypothesis: true difference in means between group after and group before is less than 0
95 percent confidence interval:
       -Inf -0.1088462
sample estimates:
 mean in group after mean in group before
                4.62                 5.12
```

Figure 4: Illinois traffic deaths: output

```r
  tibble(sim = 1:1000) %>%
    rowwise() %>%
    mutate(my_sample = list(rnorm(40, 90, 25))) %>%
    mutate(t_test = list(t.test(my_sample, mu = 100, alternative = "less"))) %>%
    mutate(p_val = t_test$p.value) %>%
    count(p_val <= 0.05)
```

```
# A tibble: 2 x 2
  `p_val <= 0.05`     n
  <lgl>           <int>
1 FALSE             210
2 TRUE              790
```

Figure 5: Power analysis

```
Rows: 24 Columns: 1
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (1): copper

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

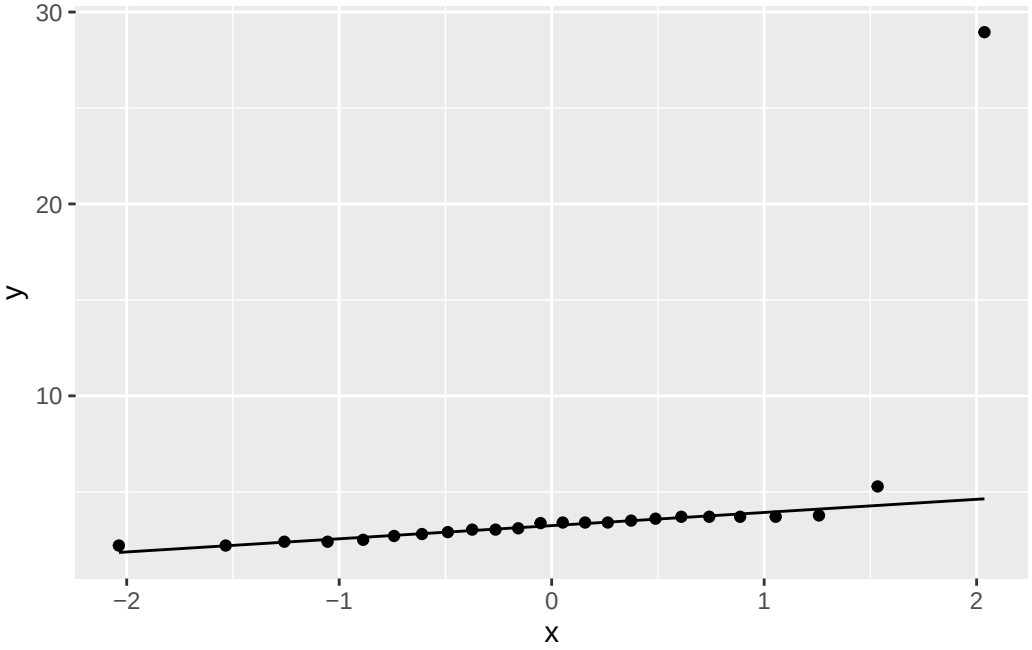

Figure 6: Normal quantile plot of copper data

```
$above_below
below above
   22     2

$p_values
  alternative     p_value
1       lower 0.000017941
2       upper 0.999998510
3   two-sided 0.000035882
```

Figure 7: Sign test for copper data

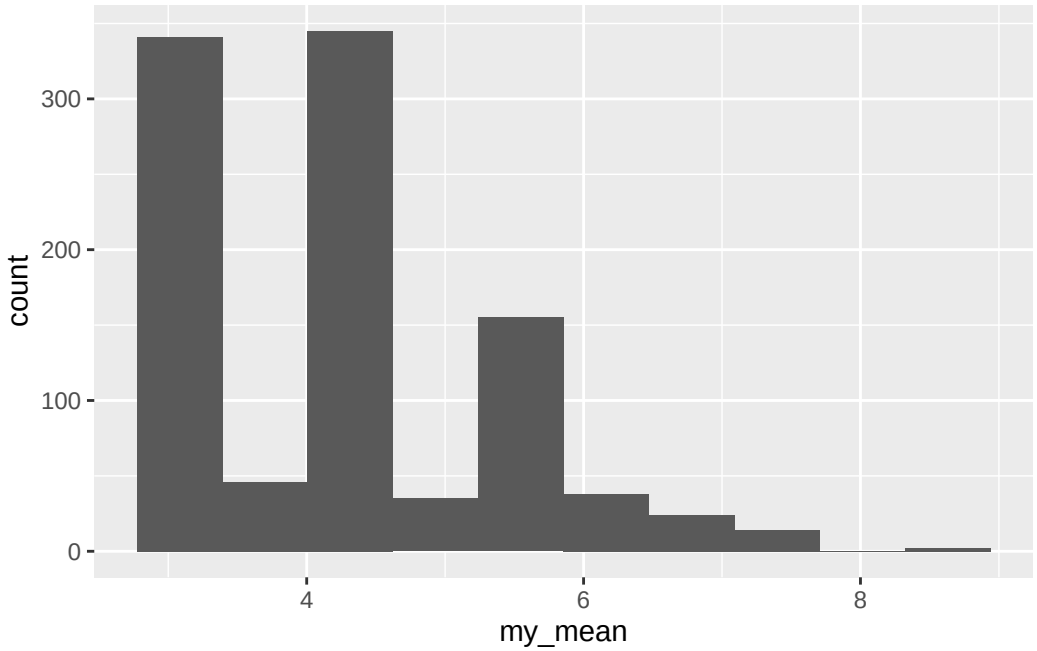

Figure 8: Bootstrap sampling distribution of sample mean for copper data

```
Rows: 39 Columns: 2
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): ration
dbl (1): weight_gain

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
  chickens
```

```
# A tibble: 39 x 2
   ration  weight_gain
   <chr>         <dbl>
 1 Ration1           4
 2 Ration2           3
 3 Ration3           6
 4 Ration1           4
 5 Ration2           4
 6 Ration3           7
 7 Ration1           7
 8 Ration2           5
 9 Ration3           7
10 Ration1           3
# i 29 more rows
```

Figure 9: Chicken weight gain data (some)

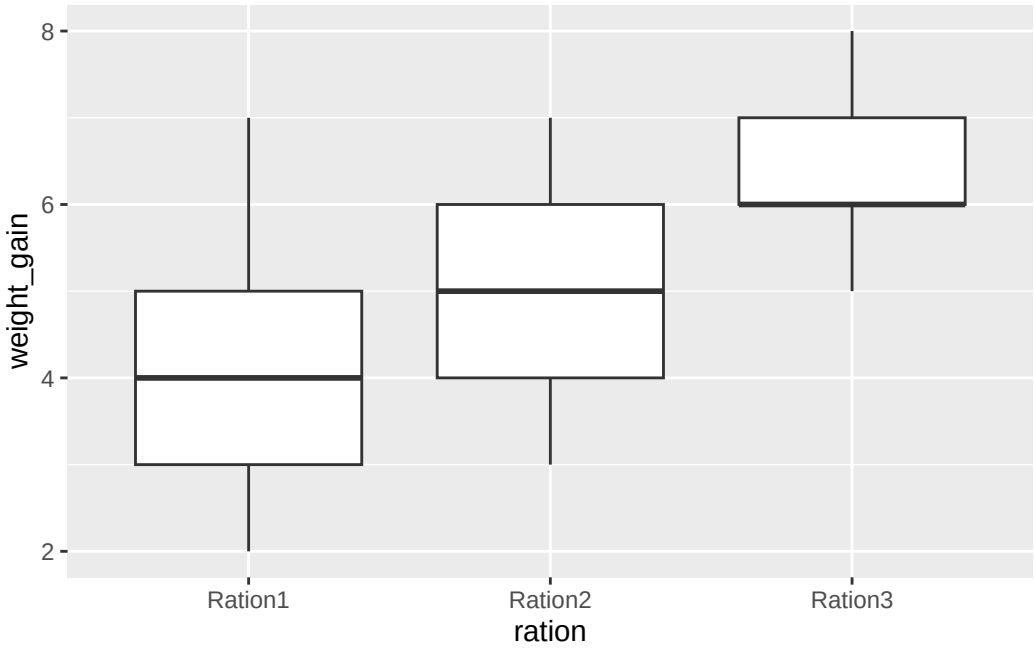

Figure 10: Boxplot of chicken weight gain data

```
chickens.1 <- aov(weight_gain ~ ration, data = chickens)
summary(chickens.1)
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)
ration       2  32.97  16.487   12.17 9.17e-05 ***
Residuals   36  48.77   1.355
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 11: Chicken weight gain analysis 1

```
oneway.test(weight_gain ~ ration, data = chickens)
```

```
    One-way analysis of means (not assuming equal variances)

data:  weight_gain and ration
F = 13.207, num df = 2.000, denom df = 23.094, p-value = 0.00015
```

Figure 12: Chicken weight gain analysis 2

```
median_test(chickens, weight_gain, ration)
```

```
$grand_median
[1] 5

$table
         above
group     above below
  Ration1     2     8
  Ration2     4     4
  Ration3    11     0

$test
       what        value
1 statistic 1.415882e+01
2        df 2.000000e+00
3   P-value 8.422685e-04
```

Figure 13: Chicken weight gain analysis 3

```
pairwise_median_test(chickens, weight_gain, ration)
```

```
# A tibble: 3 x 4
  g1      g2       p_value adj_p_value
  <chr>   <chr>      <dbl>       <dbl>
1 Ration1 Ration2 0.180        0.539
2 Ration1 Ration3 0.000415     0.00125
3 Ration2 Ration3 0.00494      0.0148
```

Figure 14: Chicken weight gain followup analysis 1

```
TukeyHSD(chickens.1)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = weight_gain ~ ration, data = chickens)

$ration
                     diff        lwr      upr     p adj
Ration2-Ration1 0.8461538 -0.2697301 1.962038 0.1670506
Ration3-Ration1 2.2307692  1.1148853 3.346653 0.0000620
Ration3-Ration2 1.3846154  0.2687315 2.500499 0.0121438
```

Figure 15: Chicken weight gain followup analysis 2

```
library(PMCMRplus)
gamesHowellTest(weight_gain ~ factor(ration), data = chickens)
```

```
    Pairwise comparisons using Games-Howell test
data: weight_gain by factor(ration)
        Ration1 Ration2
Ration2 0.23593 -
Ration3 0.00039 0.00416

P value adjustment method: none
alternative hypothesis: two.sided
```

Figure 16: Chicken weight gain followup analysis 3

```
d1
```

```
# A tibble: 3 x 3
      r     A     B
  <dbl> <dbl> <dbl>
1     1    10    21
2     2    12    19
3     3    13    22
```

Figure 17: Dataframe `d1`

```
d2
```

```
# A tibble: 6 x 3
      r treatment     y
  <dbl> <chr>     <dbl>
1     1 A            10
2     1 B            21
3     2 A            12
4     2 B            19
5     3 A            13
6     3 B            22
```

Figure 18: Dataframe `d2`

```
d3
```

```
# A tibble: 2 x 4
  malexS1 malexS2 femalexS1 femalexS2
    <dbl>   <dbl>     <dbl>     <dbl>
1    21.5    14.5      14.8      12.1
2    19.6    17.4      15.6      11.4
```

Figure 19: Dataframe `d3`

```
d4
```

```
# A tibble: 4 x 3
      r g         y
  <dbl> <chr> <dbl>
1     1 A        10
2     2 B        12
3     2 C        14
4     3 B        17
```

```
d4 %>% pivot_wider(names_from = g, values_from = y)
```

Figure 20: Dataframe `d4` and some code that uses it

```
Rows: 30 Columns: 4
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (4): taste, Acetic, H2S, Lactic

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
cheddar
```

```
# A tibble: 30 x 4
   taste Acetic   H2S Lactic
   <dbl>  <dbl> <dbl>  <dbl>
 1  12.3   4.54  3.14   0.86
 2  20.9   5.16  5.04   1.53
 3  39     5.37  5.44   1.57
 4  47.9   5.76  7.50   1.81
 5   5.6   4.66  3.81   0.99
 6  25.9   5.70  7.60   1.09
 7  37.3   5.89  8.73   1.29
 8  21.9   6.08  7.97   1.78
 9  18.1   4.90  3.85   1.29
10  21     5.24  4.17   1.58
# i 20 more rows
```

Figure 21: Cheddar cheese data

```
cheddar %>% pivot_longer(-taste, names_to = "x_name", values_to = "x_value") %>%
  ggplot(aes(x = x_value, y = taste)) + geom_point() + geom_smooth() +
  facet_wrap(~ x_name, scales = "free", ncol = 2)
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

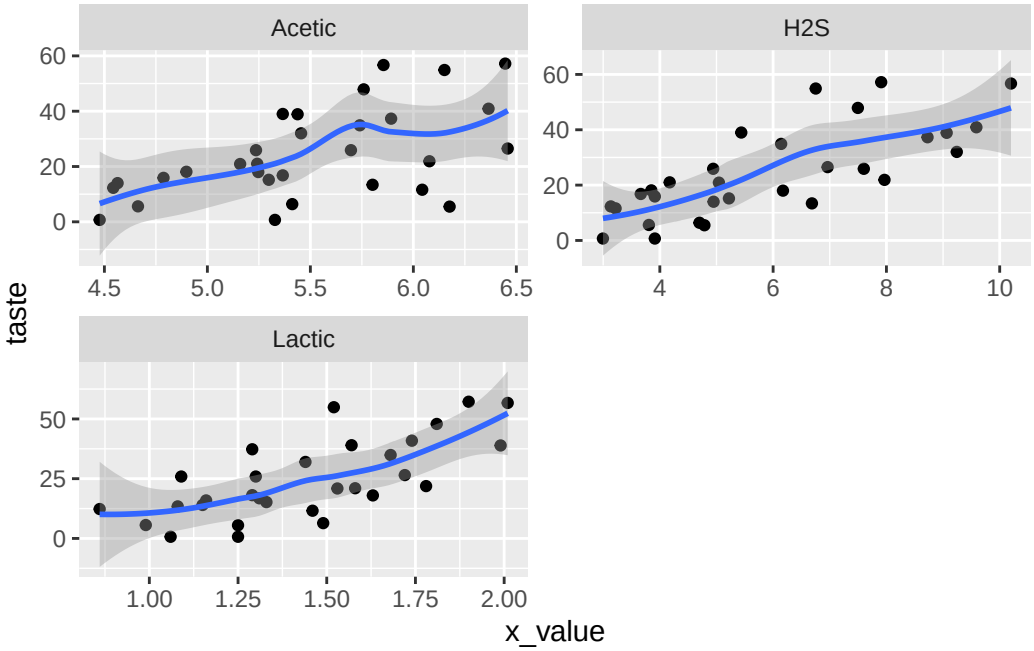

Figure 22: Cheddar scatterplots

```r
cheddar.1 <- lm(taste ~ Acetic + H2S + Lactic, data = cheddar)
summary(cheddar.1)
```

```
Call:
lm(formula = taste ~ Acetic + H2S + Lactic, data = cheddar)

Residuals:
    Min      1Q  Median      3Q     Max
-17.390  -6.612  -1.009   4.908  25.449

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -28.8768    19.7354  -1.463  0.15540
Acetic        0.3277     4.4598   0.073  0.94198
H2S           3.9118     1.2484   3.133  0.00425 **
Lactic       19.6705     8.6291   2.280  0.03108 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.13 on 26 degrees of freedom
Multiple R-squared:  0.6518,    Adjusted R-squared:  0.6116
F-statistic: 16.22 on 3 and 26 DF,  p-value: 3.81e-06
```

Figure 23: Cheddar regression model 1

```
ggplot(cheddar.1, aes(x = .fitted, y = .resid)) + geom_point()
```

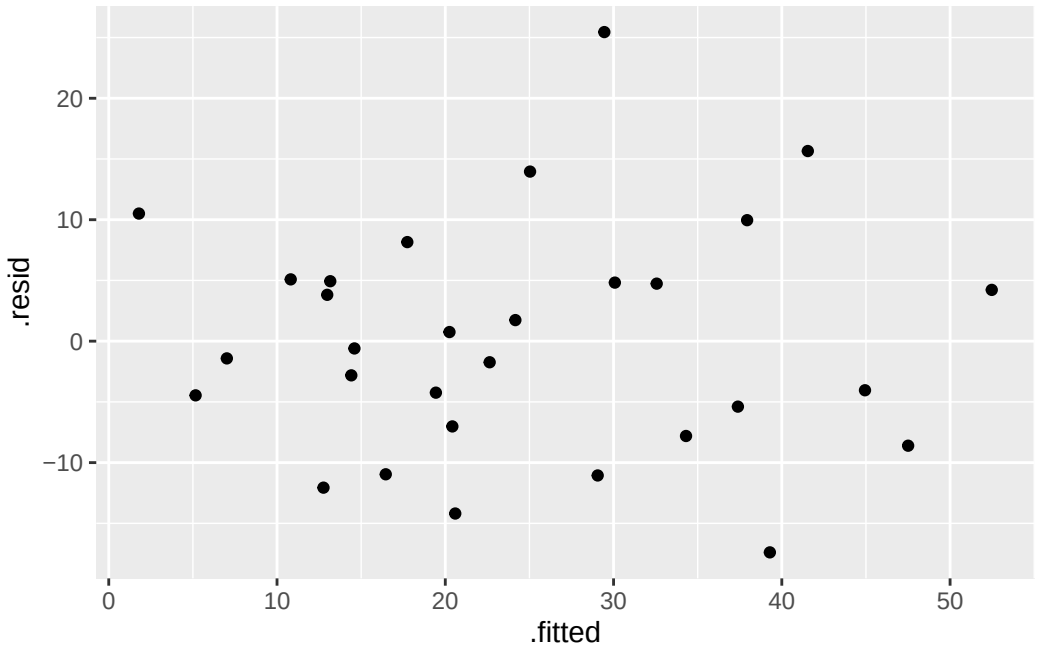

Figure 24: Cheddar residual plots 1

```
ggplot(cheddar.1, aes(sample = .resid)) + stat_qq() + stat_qq_line()
```

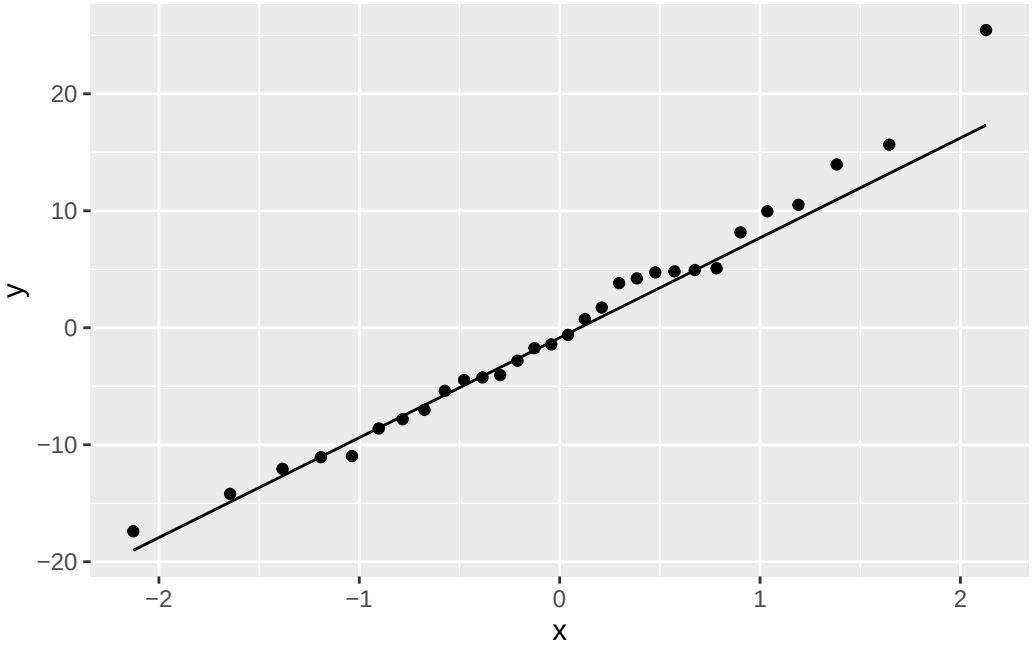

Figure 25: Cheddar residual plots 2

```
cheddar.1 %>% augment(cheddar) %>%
  pivot_longer(c(H2S, Acetic, Lactic), names_to = "x_names", values_to = "x_values") %>%
  ggplot(aes(x = x_values, y = .resid)) + geom_point() +
  facet_wrap(~ x_names, scales = "free", ncol = 2)
```

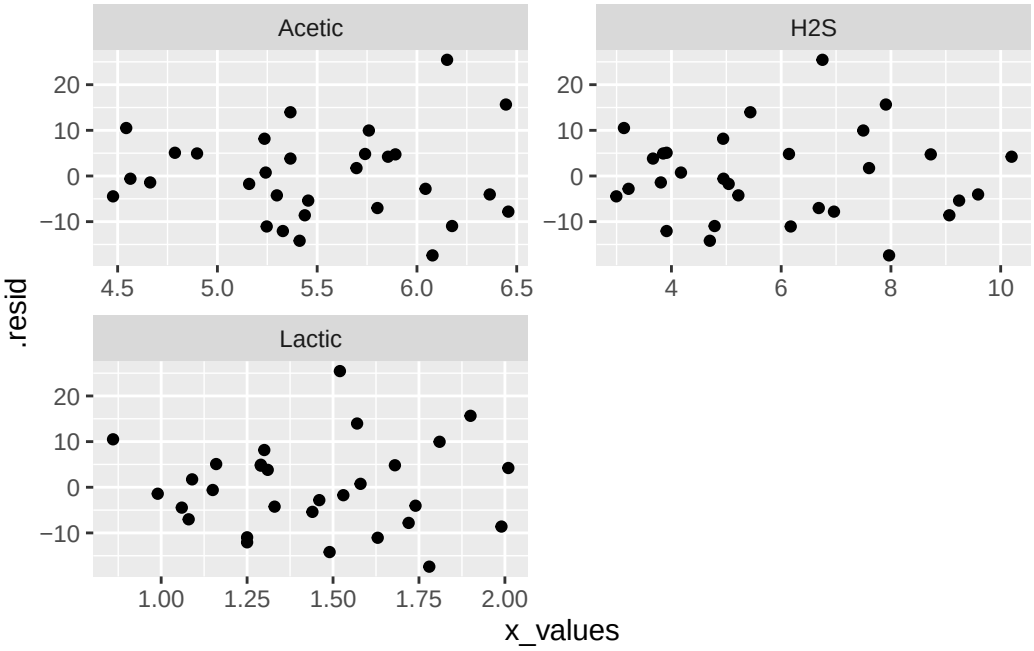

Figure 26: Cheddar residual plots 3

```r
boxcox(taste ~ Acetic + H2S + Lactic, data = cheddar)
```

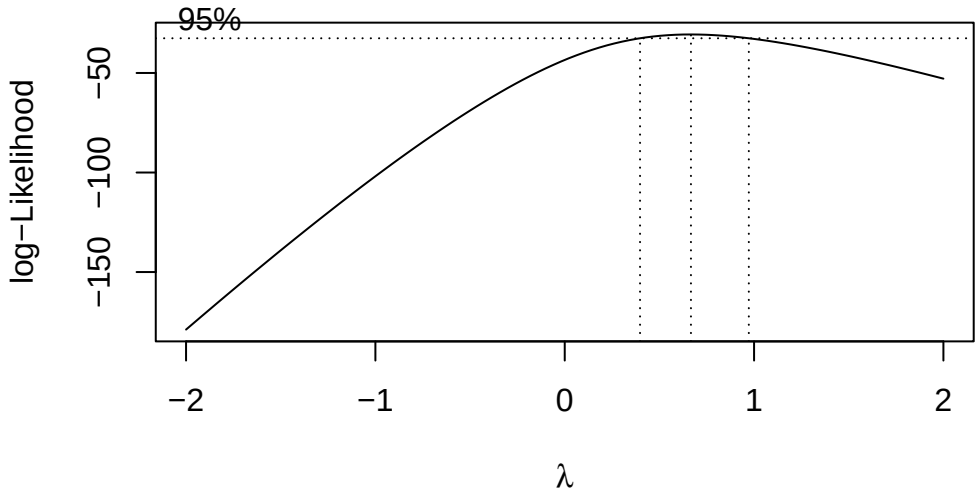

Figure 27: Cheddar further analysis

```r
d
```

```
# A tibble: 8 x 1
      x
  <dbl>
1    10
2     4
3     8
4     7
5     3
6     9
7     8
8    10
```

Figure 28: Assignment marks for a student

```
[1] 8.666667
```

Figure 29: Result of running function