

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()  
)
```

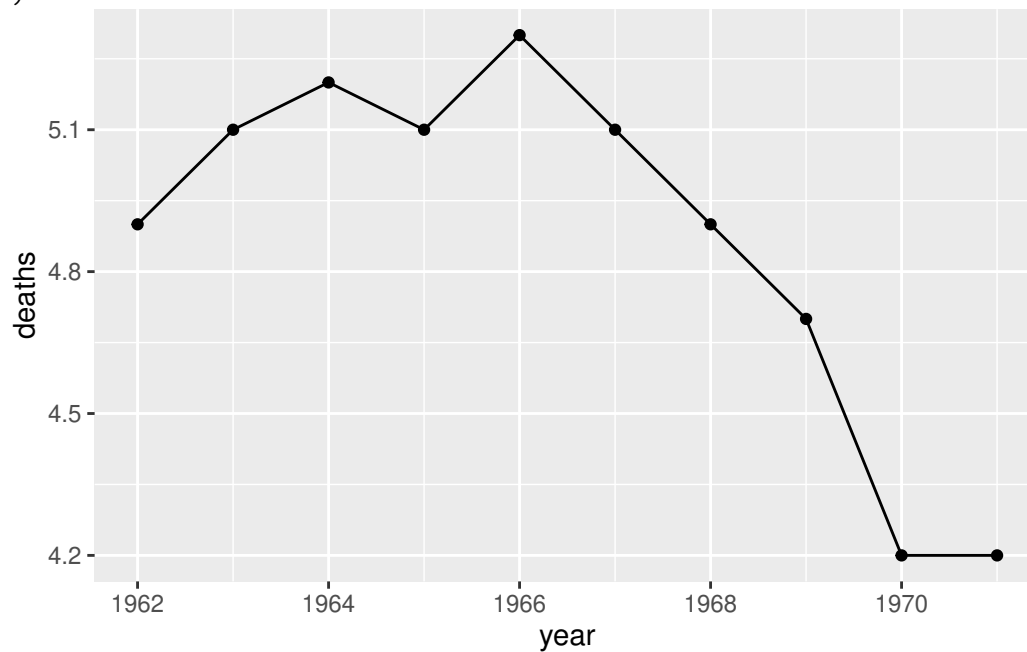


Figure 3: Illinois traffic deaths: graph

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

```

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
  <lg1>              <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.

```

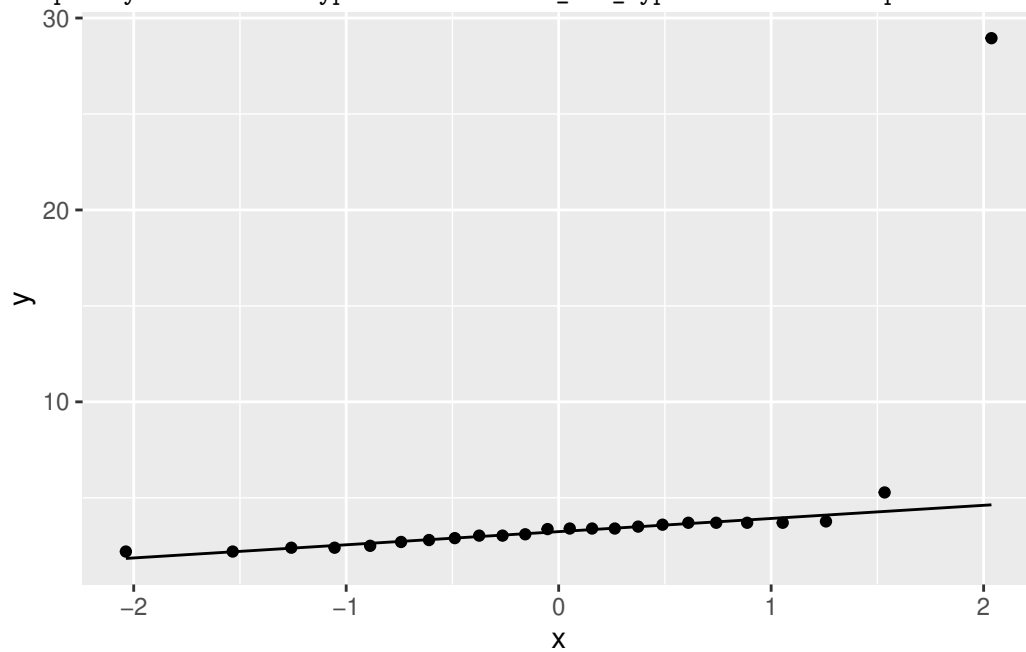


Figure 6: Normal quantile plot of copper data

```
$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

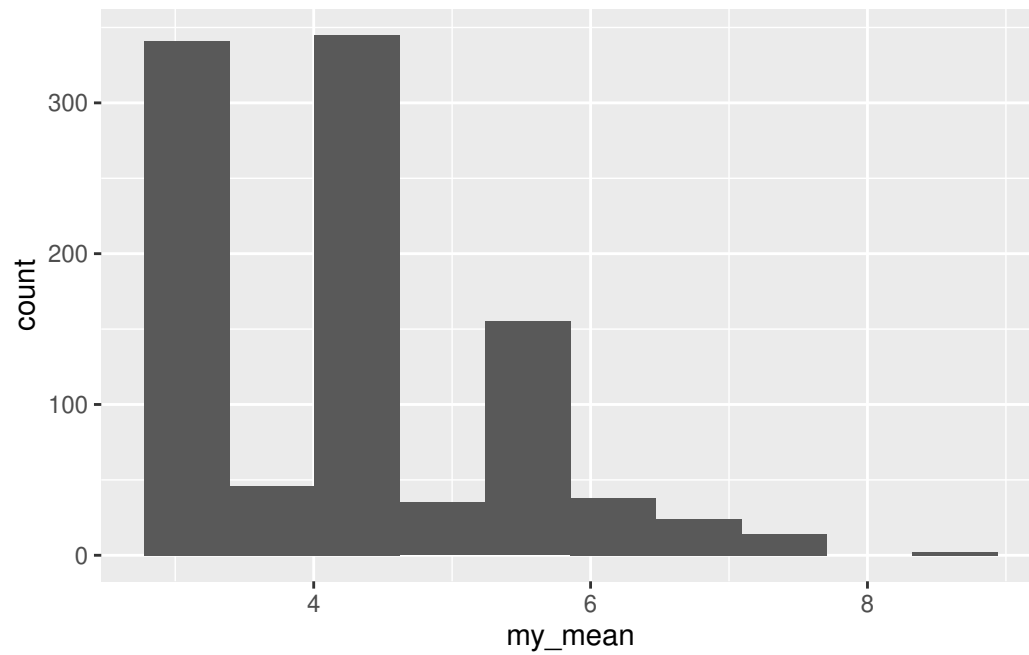


Figure 8: Bootstrap sampling distribution of sample mean for copper data

```
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)

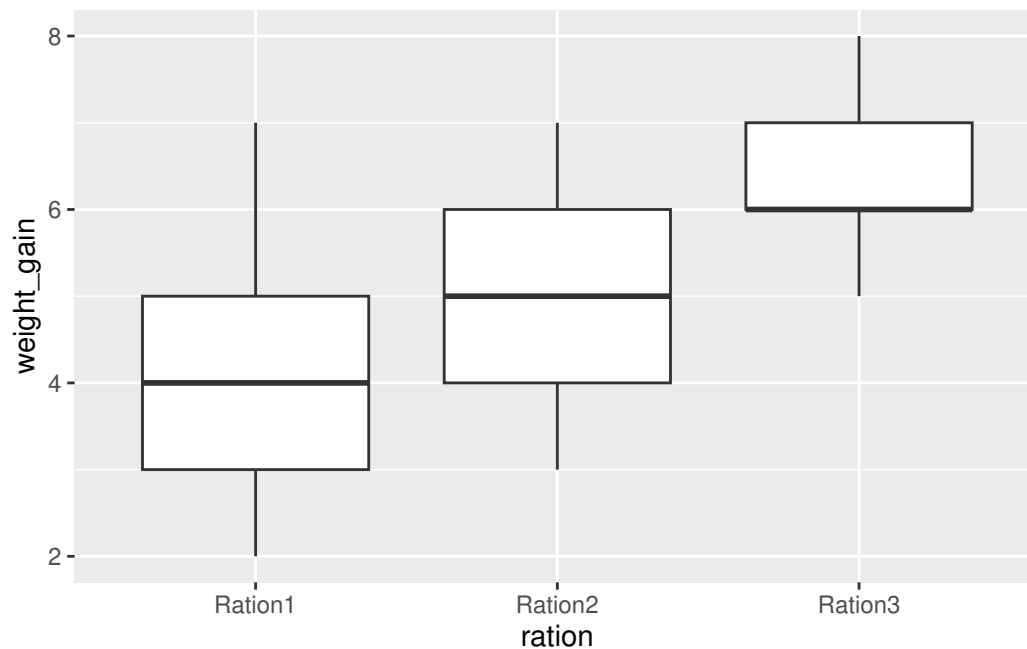


Figure 10: Boxplot of chicken weight gain data

```
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)

$racion
      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'

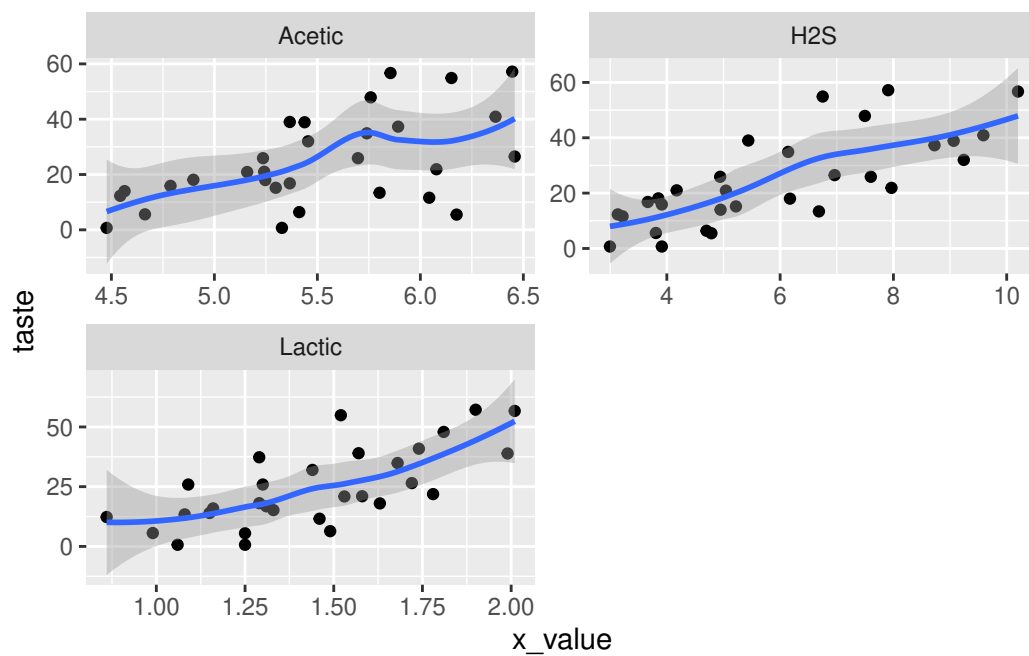


Figure 22: Cheddar scatterplots

```
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()
```

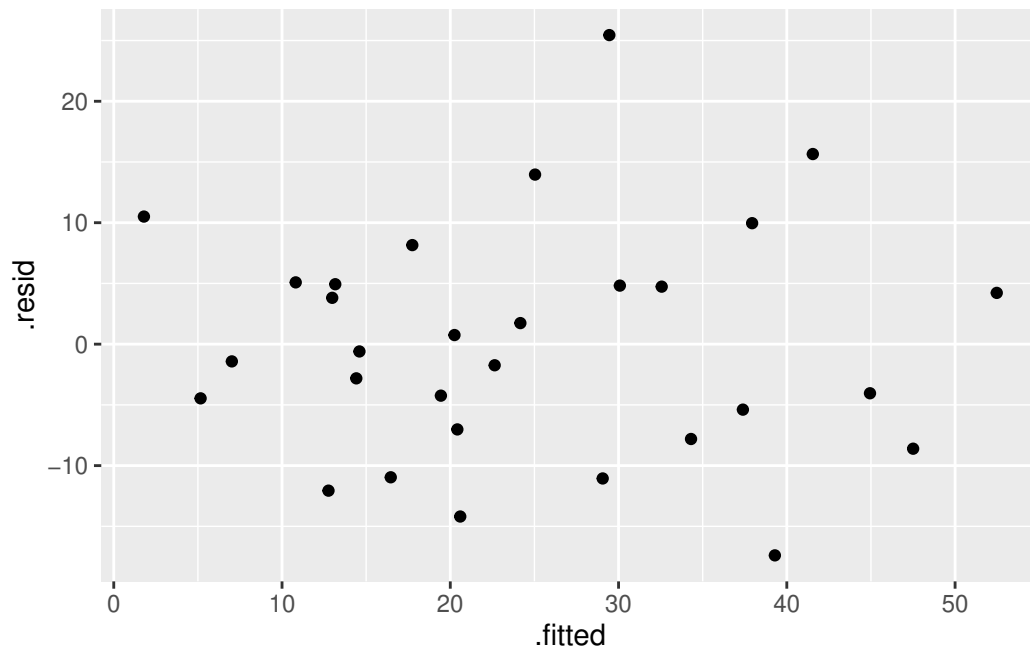


Figure 24: Cheddar residual plots 1

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

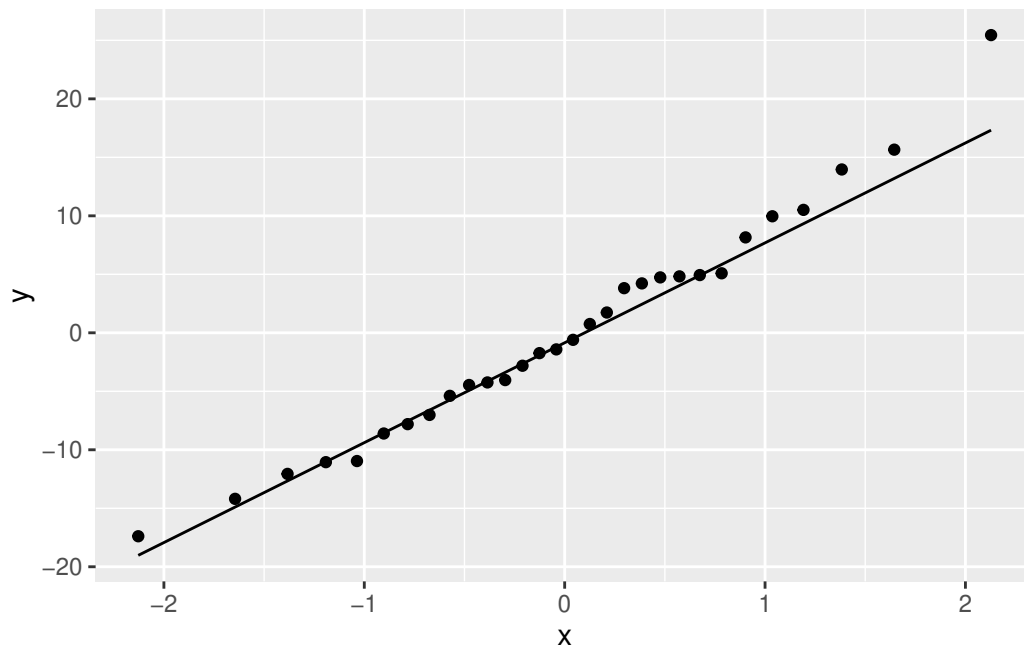


Figure 25: Cheddar residual plots 2

```
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)
```

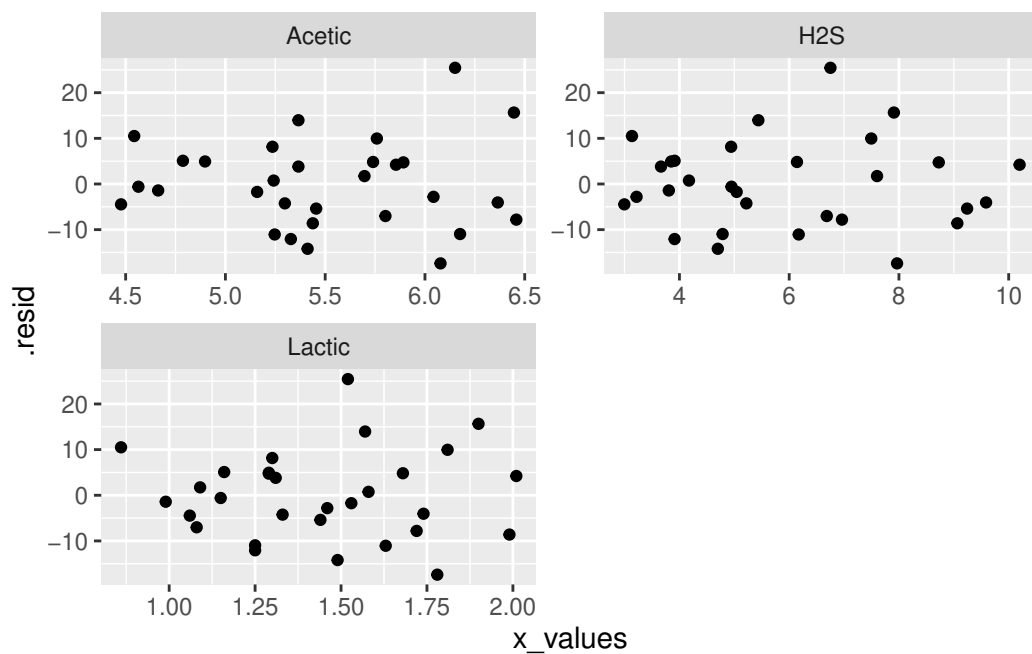


Figure 26: Cheddar residual plots 3


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

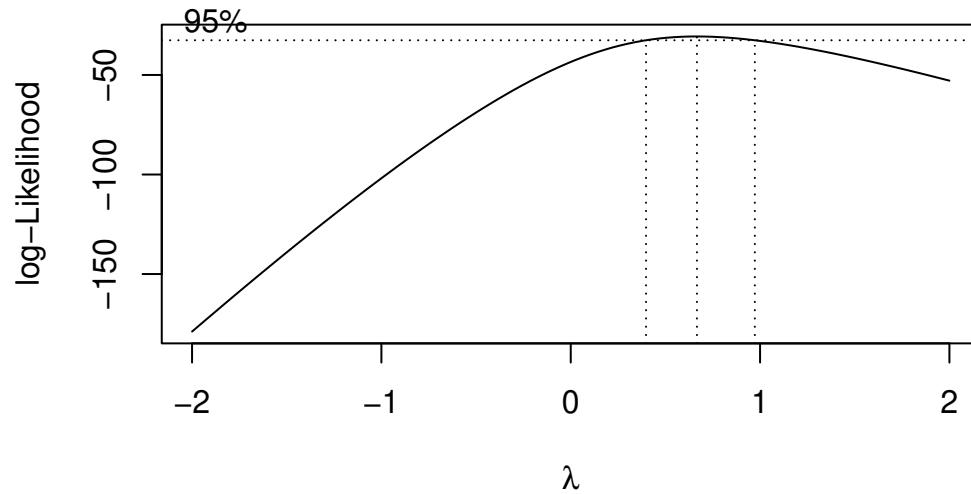


Figure 27: Cheddar further analysis

```
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