Booklet of Code and Output

for

STAD29/STA 1007 Midterm Exam

List of Figures in this document by page:

# List of Figures

```r
library(MASS)
library(tidyverse)

## -- Attaching packages ---------------------------------
tidyverse 1.2.1 --
## ✓ ggplot2 2.2.1.9000    ✓ purrr  0.2.4
## ✓ tibble  1.4.2          ✓ dplyr  0.7.4
## ✓ tidyr   0.8.0          ✓ stringr 1.3.0
## ✓ readr   1.1.1          ✓ forcats 0.3.0
## -- Conflicts ------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()

library(broom)
library(survival)
library(survminer)

## Loading required package: ggpubr
## Loading required package: magrittr
##
## Attaching package: 'magrittr'
## The following object is masked from 'package:purrr':
##
##     set_names
## The following object is masked from 'package:tidyr':
##
##     extract
```

Figure 1: Packages

```
infection=read_tsv("infectionrisk.txt")

## Parsed with column specification:
## cols(
##   ID = col_integer(),
##   Stay = col_double(),
##   Age = col_double(),
##   InfctRsk = col_double(),
##   Culture = col_double(),
##   Xray = col_double(),
##   Beds = col_integer(),
##   MedSchool = col_integer(),
##   Region = col_integer(),
##   Census = col_integer(),
##   Nurses = col_integer(),
##   Facilities = col_double()
## )

infection

## # A tibble: 113 x 12
##        ID  Stay   Age InfctRsk Culture  Xray  Beds MedSchool Region Census
##     <int> <dbl> <dbl>    <dbl>   <dbl> <dbl> <int>     <int>  <int>  <int>
##  1      1  7.13  55.7     4.10    9.00  39.6   279         2      4    207
##  2      2  8.82  58.2     1.60    3.80  51.7    80         2      2     51
##  3      3  8.34  56.9     2.70    8.10  74.0   107         2      3     82
##  4      4  8.95  53.7     5.60   18.9  123     147         2      4     53
##  5      5 11.2   56.5     5.70   34.5   88.9   180         2      1    134
##  6      6  9.76  50.9     5.10   21.9   97.0   150         2      2    147
##  7      7  9.68  57.8     4.60   16.7   79.0   186         2      3    151
##  8      8 11.2   45.7     5.40   60.5   85.8   640         1      2    399
##  9      9  8.67  48.2     4.30   24.4   90.8   182         2      3    130
## 10     10  8.84  56.3     6.30   29.6   82.6    85         2      1     59
## # ... with 103 more rows, and 2 more variables: Nurses <int>,
## #   Facilities <dbl>
```

Figure 2: Hospital infection risk data (some)

```
infection = infection %>% mutate(Region=factor(Region))
inf.1=lm(InfctRsk~Stay+Xray+Region,data=infection)
summary(inf.1)

##
## Call:
## lm(formula = InfctRsk ~ Stay + Xray + Region, data = infection)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.75483 -0.64146  0.00862  0.67124  2.44950
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.802903   0.775573  -1.035 0.302892
## Stay         0.349288   0.063845   5.471 2.97e-07 ***
## Xray         0.019663   0.005762   3.413 0.000909 ***
## Region2      0.178873   0.290077   0.617 0.538782
## Region3      0.043021   0.297064   0.145 0.885124
## Region4      0.832871   0.381718   2.182 0.031304 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.068 on 107 degrees of freedom
## Multiple R-squared:  0.3938,Adjusted R-squared:  0.3655
## F-statistic:  13.9 on 5 and 107 DF,  p-value: 1.839e-10
```

Figure 3: Regression for predicting infection risk

```
drop1(inf.1,test="F")

## Single term deletions
##
## Model:
## InfctRsk ~ Stay + Xray + Region
##        Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>              122.07 20.727
## Stay    1    34.147 156.22 46.598 29.9305 2.968e-07 ***
## Xray    1    13.287 135.36 30.402 11.6464 0.0009092 ***
## Region  3     7.334 129.41 21.320  2.1428 0.0991208 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 4: Drop-1 output from regression

```
inf.2=update(inf.1,.~.-Region)
new=tibble(Stay=15,Xray=70,Region=1)
p=predict(inf.2,new,interval="p")
cbind(new,p)

##   Stay Xray Region      fit     lwr      upr
## 1   15   70      1 5.702933 3.44029 7.965575
```

Figure 5: Another model, and predictions

```
flu=read_table("flu-shots.txt")

## Parsed with column specification:
## cols(
##   shot = col_double(),
##   age = col_double(),
##   awareness = col_double()
## )

flu

## # A tibble: 50 x 3
##     shot   age awareness
##    <dbl> <dbl>     <dbl>
##  1 0      38.0      40.0
##  2 1.00   52.0      60.0
##  3 0      41.0      36.0
##  4 1.00   46.0      59.0
##  5 1.00   41.0      70.0
##  6 0      43.0      49.0
##  7 1.00   57.0      59.0
##  8 0      34.0      50.0
##  9 0      31.0      48.0
## 10 1.00   49.0      59.0
## # ... with 40 more rows
```

Figure 6: Flu shot data (some)

```r
shot.1=glm(factor(shot)~age+awareness, family="binomial", data=flu)
summary(shot.1)

##
## Call:
## glm(formula = factor(shot) ~ age + awareness, family = "binomial",
##     data = flu)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5522  -0.2962  -0.1124   0.4208   2.3244
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -21.58458    6.41824  -3.363 0.000771 ***
## age           0.22178    0.07436   2.983 0.002858 **
## awareness     0.20351    0.06273   3.244 0.001178 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 68.029  on 49  degrees of freedom
## Residual deviance: 32.416  on 47  degrees of freedom
## AIC: 38.416
##
## Number of Fisher Scoring iterations: 6
```

Figure 7: Logistic regression

```r
flu %>%
    summarize(age_q1=quantile(age,0.25),
              age_q3=quantile(age,0.75),
              awareness_q1=quantile(awareness,0.25),
              awareness_q3=quantile(awareness,0.75))

## # A tibble: 1 x 4
##   age_q1 age_q3 awareness_q1 awareness_q3
##    <dbl>  <dbl>        <dbl>        <dbl>
## 1   40.2   53.0         43.2         59.0
```

Figure 8: Quartiles for age and awareness

```r
kids=read_csv("kids.csv")

## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   pupilid = col_double(),
##   ks2score = col_double(),
##   ks3score = col_double(),
##   ks4score = col_double(),
##   IDACI_n = col_double(),
##   weighting = col_double()
## )
## See spec(...) for full column specifications.

kids = kids %>% select(k3en,gender,sec,ks2stand)
kids

## # A tibble: 15,770 x 4
##     k3en gender   sec ks2stand
##    <int>  <int> <int>    <int>
##  1     3      0     2      -24
##  2     3      0     8       NA
##  3     3      1    NA       NA
##  4     3      0     2      -21
##  5     3      1    NA      -24
##  6     3      1    NA      -24
##  7     3      1    NA      -24
##  8     3      1    NA      -24
##  9     3      1     8       NA
## 10     3      1     2      -24
## # ... with 15,760 more rows
```

(Note: 0 is male and 1 is female)

Figure 9: LSYPE data, some, selected variables

```
summary(kids)

##       k3en           gender             sec           ks2stand
##  Min.   :3.000   Min.   :0.0000   Min.   :1.000   Min.   :-
24.0000
##  1st Qu.:4.000   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.: -
7.0000
##  Median :5.000   Median :0.0000   Median :4.000   Median :  0.0000
##  Mean   :5.067   Mean   :0.4912   Mean   :4.114   Mean   :  0.0119
##  3rd Qu.:6.000   3rd Qu.:1.0000   3rd Qu.:6.000   3rd Qu.:  7.0000
##  Max.   :7.000   Max.   :1.0000   Max.   :8.000   Max.   : 39.0000
##  NA's   :1307    NA's   :339      NA's   :2941    NA's   :1469
```

Figure 10: Summary of data

```
kids = kids %>%
    filter(!is.na(k3en),
           !is.na(gender),
           !is.na(sec),
           !is.na(ks2stand))
summary(kids)

##       k3en           gender             sec           ks2stand
##  Min.   :3.000   Min.   :0.0000   Min.   :1.000   Min.   :-
24.0000
##  1st Qu.:5.000   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.: -
6.0000
##  Median :5.000   Median :0.0000   Median :4.000   Median :  1.0000
##  Mean   :5.139   Mean   :0.4889   Mean   :4.119   Mean   :  0.6265
##  3rd Qu.:6.000   3rd Qu.:1.0000   3rd Qu.:6.000   3rd Qu.:  7.0000
##  Max.   :7.000   Max.   :1.0000   Max.   :8.000   Max.   : 39.0000
```

Figure 11: Doing something with our variables

```
en3.1=polr(en3~gender+sec+ks2stand,data=kids)
drop1(en3.1,test="Chisq")

## Single term deletions
##
## Model:
## en3 ~ gender + sec + ks2stand
##          Df   AIC    LRT  Pr(>Chi)
## <none>      22208
## gender    1 22911  704.4 < 2.2e-16 ***
## sec       1 22496  289.4 < 2.2e-16 ***
## ks2stand  1 30381 8174.3 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 12: Model-fitting

Probabilities of obtaining a Key Stage 3 English grade of 3, 4, 5, 6 or 7 from values of explanatory variables as shown. Code to obtain the predictions is not shown:

```
cbind(new,round(p,3))

##    gender sec ks2stand     3     4     5     6     7
## 1       0   1       -7 0.083 0.323 0.537 0.053 0.004
## 2       0   1        0 0.018 0.106 0.651 0.206 0.018
## 3       0   1        7 0.004 0.025 0.390 0.499 0.082
## 4       0   6       -7 0.162 0.432 0.378 0.026 0.002
## 5       0   6        0 0.039 0.195 0.647 0.111 0.009
## 6       0   6        7 0.008 0.052 0.547 0.353 0.040
## 7       1   1       -7 0.032 0.168 0.658 0.131 0.011
## 8       1   1        0 0.007 0.043 0.508 0.394 0.049
## 9       1   1        7 0.001 0.009 0.197 0.595 0.197
## 10      1   6       -7 0.066 0.282 0.580 0.067 0.005
## 11      1   6        0 0.014 0.086 0.629 0.247 0.023
## 12      1   6        7 0.003 0.020 0.337 0.537 0.103
```

Note that `round` rounds the variable (given first) to the given number of decimals (second).

Figure 13: Predictions for LSYPE English grade

```
unemp=read_csv("unemployment.csv")

## Parsed with column specification:
## cols(
##   spell = col_integer(),
##   event = col_integer(),
##   ui = col_integer(),
##   logwage = col_double(),
##   work_area = col_character()
## )

unemp

## # A tibble: 1,957 x 5
##    spell event    ui logwage work_area
##    <int> <int> <int>   <dbl> <chr>
##  1     1     1     0    6.41 mining
##  2     3     0     1    5.85 mining
##  3     2     1     0    6.57 mining
##  4     3     0     1    5.76 mining
##  5     2     0     1    5.38 mining
##  6     5     0     1    5.56 mining
##  7     7     0     1    6.11 mining
##  8     4     0     1    6.34 mining
##  9     3     0     1    5.99 mining
## 10     8     0     0    5.83 mining
## # ... with 1,947 more rows
```

Figure 14: Unemployment data (some)

```
y=with(unemp,Surv(spell,event))
y[1:20]

##  [1]  1   3+  2   3+  2+  5+  7+  4+  3+  8+  2  13  11+ 12+  1  17+  4+
## [18]  7+  7+  5
```

Figure 15: Construction of response variable and display of first 20 values

```r
y.1=coxph(y~ui+logwage+work_area,data=unemp)
summary(y.1)

## Call:
## coxph(formula = y ~ ui + logwage + work_area, data = unemp)
##
##   n= 1957, number of events= 658
##
##                         coef exp(coef) se(coef)       z Pr(>|z|)
## ui                  -0.99193   0.37086  0.08275 -11.987  < 2e-16 ***
## logwage              0.44326   1.55778  0.06979   6.352 2.13e-10 ***
## work_areafire        0.53674   1.71041  0.14922   3.597 0.000322 ***
## work_areamining     -0.13158   0.87671  0.21709  -0.606 0.544450
## work_areapubadmin -0.24263   0.78456  0.41874  -0.579 0.562301
## work_areaservices  0.34281   1.40889  0.11727   2.923 0.003465 **
## work_areatrade     0.18117   1.19861  0.11782   1.538 0.124133
## work_areatransp   -0.09024   0.91371  0.15395  -0.586 0.557740
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                   exp(coef) exp(-coef) lower .95 upper .95
## ui                   0.3709     2.6964    0.3153    0.4362
## logwage              1.5578     0.6419    1.3586    1.7861
## work_areafire        1.7104     0.5847    1.2767    2.2915
## work_areamining      0.8767     1.1406    0.5729    1.3417
## work_areapubadmin    0.7846     1.2746    0.3453    1.7826
## work_areaservices    1.4089     0.7098    1.1196    1.7730
## work_areatrade       1.1986     0.8343    0.9515    1.5100
## work_areatransp      0.9137     1.0944    0.6757    1.2355
##
## Concordance= 0.697  (se = 0.014 )
## Rsquare= 0.09   (max possible= 0.99 )
## Likelihood ratio test= 184.1  on 8 df,   p=0
## Wald test            = 185  on 8 df,   p=0
## Score (logrank) test = 193.1  on 8 df,   p=0

drop1(y.1,test="Chisq")

## Single term deletions
##
## Model:
## y ~ ui + logwage + work_area
##           Df    AIC     LRT  Pr(>Chi)
## <none>       8815.1
## ui         1 8959.4 146.269 < 2.2e-16 ***
## logwage    1 8852.6  39.511 3.261e-10 ***
## work_area  6 8828.7  25.644 0.0002594 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 16: Cox model

```
unemp %>% summarize(med=median(logwage))

## # A tibble: 1 x 1
##     med
##   <dbl>
## 1  5.69

work_areas = unemp %>% distinct(work_area) %>% pull(work_area)
work_areas

## [1] "mining"   "constr"   "transp"   "trade"    "fire"     "services"
## [7] "pubadmin"

unemp_new=crossing(logwage=5.69,ui=1,work_area=work_areas)
unemp_new

## # A tibble: 7 x 3
##   logwage    ui work_area
##     <dbl> <dbl> <chr>
## 1    5.69  1.00 constr
## 2    5.69  1.00 fire
## 3    5.69  1.00 mining
## 4    5.69  1.00 pubadmin
## 5    5.69  1.00 services
## 6    5.69  1.00 trade
## 7    5.69  1.00 transp

s=survfit(y.1,unemp_new,data=unemp)
```

Figure 17: Predictions for job type

```r
rods=read_csv("rodmold.csv")

## Parsed with column specification:
## cols(
##   temperature = col_integer(),
##   pressure = col_integer(),
##   batch = col_integer(),
##   extrusion_rate = col_double()
## )

rods = rods %>% mutate(pressure=factor(pressure),
                       temperature=factor(temperature))
rods

## # A tibble: 12 x 4
##    temperature pressure batch extrusion_rate
##    <fct>       <fct>    <int>          <dbl>
##  1 200         40           1           1.35
##  2 200         40           2           1.31
##  3 200         40           3           1.40
##  4 200         60           1           1.74
##  5 200         60           2           1.67
##  6 200         60           3           1.86
##  7 300         40           1           2.48
##  8 300         40           2           2.29
##  9 300         40           3           2.14
## 10 300         60           1           3.63
## 11 300         60           2           3.30
## 12 300         60           3           3.27
```

Figure 18: Rod extrusion data

```r
extr.1=aov(extrusion_rate~temperature*pressure,data=rods)
summary(extr.1)

##                      Df Sum Sq Mean Sq F value   Pr(>F)
## temperature           1  5.044   5.044  251.57 2.50e-07 ***
## pressure              1  1.687   1.687   84.17 1.61e-05 ***
## temperature:pressure  1  0.361   0.361   17.98  0.00284 **
## Residuals             8  0.160   0.020
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 19: Analysis of variance for rod extrusion data

Extrusion rate means for pressure and temperature combinations

```
rods %>% group_by(temperature,pressure) %>%
    summarize(m=mean(extrusion_rate))

## # A tibble: 4 x 3
## # Groups:   temperature [?]
##   temperature pressure     m
##   <fct>       <fct>    <dbl>
## 1 200         40        1.35
## 2 200         60        1.76
## 3 300         40        2.30
## 4 300         60        3.40
```

Tukey:

```
TukeyHSD(extr.1)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = extrusion_rate ~ temperature * pressure, data = rods)
##
## $temperature
##             diff      lwr      upr p adj
## 300-200 1.296667 1.108147 1.485186 2e-07
##
## $pressure
##       diff       lwr       upr    p adj
## 60-40 0.75 0.5614803 0.9385197 1.61e-05
##
## $`temperature:pressure`
##                     diff         lwr       upr     p adj
## 300:40-200:40  0.9500000  0.57976231  1.320238 0.0001661
## 200:60-200:40  0.4033333  0.03309564  0.773571 0.0334993
## 300:60-200:40  2.0466667  1.67642898  2.416904 0.0000005
## 200:60-300:40 -0.5466667 -0.91690436 -0.176429 0.0064699
## 300:60-300:40  1.0966667  0.72642898  1.466904 0.0000585
## 300:60-200:60  1.6433333  1.27309564  2.013571 0.0000028
```

Figure 20: Tukey for rod extrusion data

```
pval=function(x) {
    extr.2=aov(extrusion_rate~pressure,data=x)
    extr.3=glance(extr.2)
    extr.3$p.value
}
rods %>%
    group_by(temperature) %>%
    nest() %>%
    mutate(p_value=map_dbl(data,pval))

## # A tibble: 2 x 3
##   temperature data              p_value
##   <fct>       <list>              <dbl>
## 1 200         <tibble [6 x 3]> 0.00276
## 2 300         <tibble [6 x 3]> 0.00194
```

Figure 21: Further analysis of rod extrusion data

```
ggsurvplot(s,conf.int=F)
```

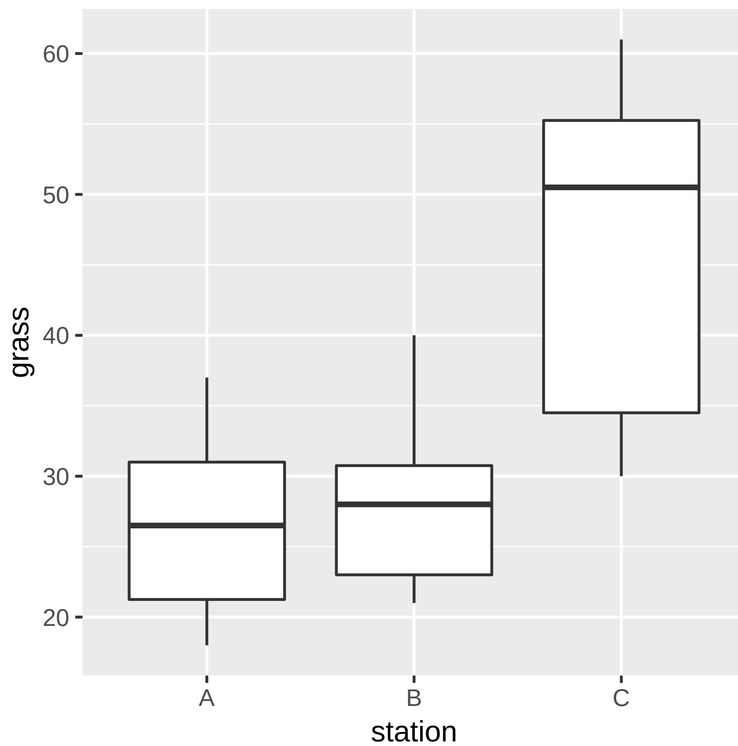


Figure 22: Plot of predictions for job type

```r
ggplot(rods,aes(y=extrusion_rate,x=temperature,
  fill=pressure))+geom_boxplot()
```

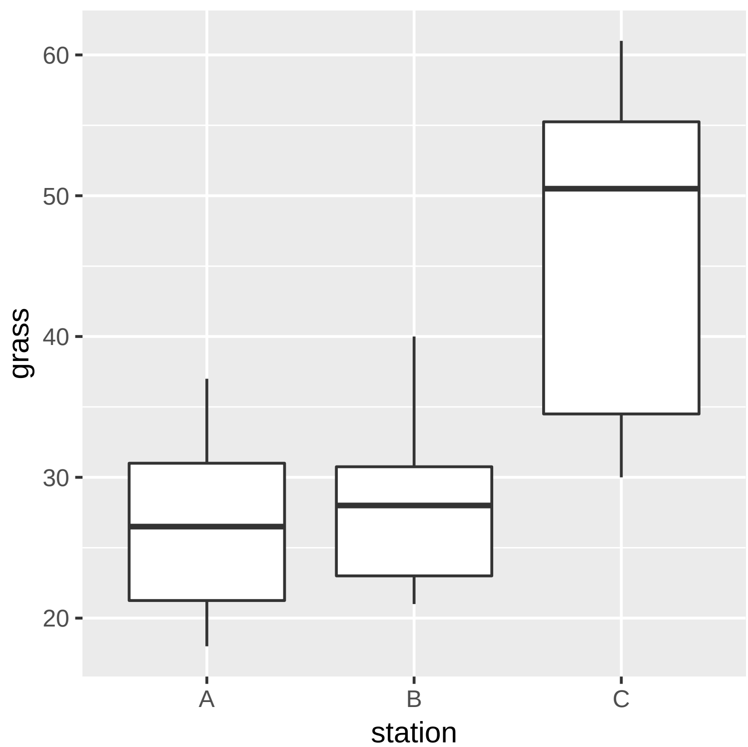


Figure 23: Grouped boxplot for rod extrusion data

```
rods.mean = rods %>% group_by(temperature,pressure) %>%
  summarize(m=mean(extrusion_rate))
rods.mean
```

```
## # A tibble: 4 x 3
## # Groups:   temperature [?]
##   temperature pressure     m
##   <fct>       <fct>    <dbl>
## 1 200         40        1.35
## 2 200         60        1.76
## 3 300         40        2.30
## 4 300         60        3.40
```

```
ggplot(rods.mean,aes(y=m,x=temperature,colour=pressure,group=pressure))+
  geom_point()+geom_line()
```

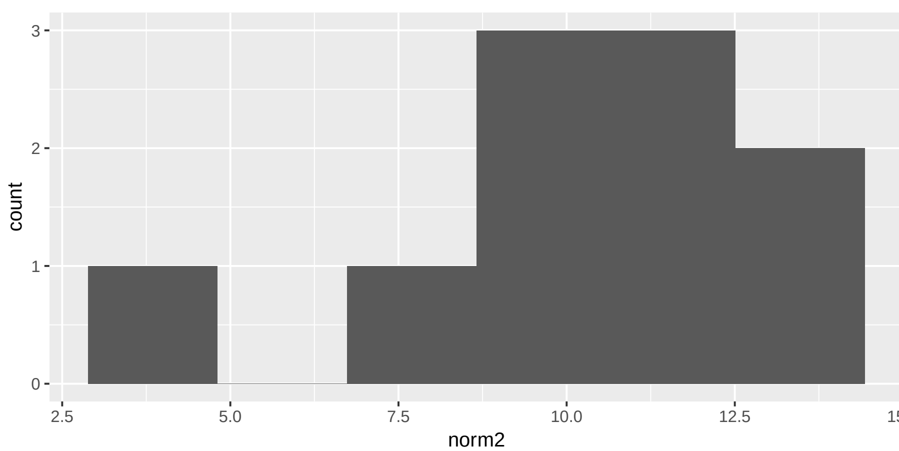


Figure 24: Interaction plot for rod extrusion data