00D7×

Booklet of Code and Output
for
STAD29/STA 1007 Midterm Exam

List of Figures in this document by page:

List of Figures

Note that Figures 33, 34 and 35 are at the *end* of this booklet, because they are printed in colour.

```
library(tidyverse)

## -- Attaching packages ---------------------------------
tidyverse 1.2.1 --
## v ggplot2 3.1.1          v purrr  0.3.2
## v tibble  2.1.1          v dplyr  0.8.0.1
## v tidyr  0.8.3.9000    v stringr 1.4.0
## v readr  1.3.1          v forcats 0.3.0
## Warning: package 'ggplot2' was built under R version 3.5.3
## Warning: package 'tibble' was built under R version 3.5.3
## Warning: package 'tidyr' was built under R version 3.5.3
## Warning: package 'readr' was built under R version 3.5.2
## Warning: package 'purrr' was built under R version 3.5.3
## Warning: package 'dplyr' was built under R version 3.5.2
## Warning: package 'stringr' was built under R version 3.5.2
## Warning: package 'forcats' was built under R version 3.5.1
## -- Conflicts ------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(broom)

## Warning: package 'broom' was built under R version 3.5.2

library(MASS)

##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##    select

library(survival)
library(survminer)

## Warning: package 'survminer' was built under R version 3.5.1
## Loading required package: ggpubr
## Warning: package 'ggpubr' was built under R version 3.5.1
## Loading required package: magrittr
## Warning: package 'magrittr' was built under R version 3.5.1
##
## 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

```
reasoning=read.csv("reasoning.csv",header=T)
reasoning
```

```
##    row piano singing computer none
## 1    1     2       1        0    5
## 2    2     5      -1        1   -1
## 3    3     7       0        1    7
## 4    4    -2       1       -3    0
## 5    5     2      -4       -2    4
## 6    6     7       0        4    0
## 7    7     4       0       -1    2
## 8    8     1       1        2    1
## 9    9     0       0        4   -6
## 10  10     7      -1        2    0
```

Figure 2: Reasoning data

```
reasoning %>%
  gather(lesson,changescore,piano:none) -> reasoning2
```

Figure 3: Processing, part 1

```
reasoning2 %>%
  group_by(lesson) %>%
  summarize(count=n(),m=mean(changescore))
```

Figure 4: Processing, part 2

```
reasoning2 %>%
  ggplot(aes(x=lesson,y=changescore))+geom_boxplot()
```

Figure 5: Processing, part 3

```r
mercury=read.csv("mercury.csv",header=T)
str(mercury)

## 'data.frame': 38 obs. of  4 variables:
##  $ mercury   : int  1330 250 450 160 720 810 710 510 1000 150 ...
##  $ alkalinity: num  2.5 19.6 5.2 71.4 26.4 4.8 6.6 16.5 7.1 83.7 ...
##  $ calcium   : num  2.9 4.5 2.8 55.2 9.2 4.6 2.7 13.8 5.2 66.5 ...
##  $ pH        : num  4.6 7.3 5.4 8.1 5.8 6.4 5.4 7.2 5.8 8.2 ...
```

Figure 6: Fish mercury data (structure)

```
mercury %>%
  gather(xname,x,alkalinity:pH) %>%
  ggplot(aes(x=x,y=mercury))+geom_point()+geom_smooth()+
    facet_wrap(~xname,scales="free",ncol=2)

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

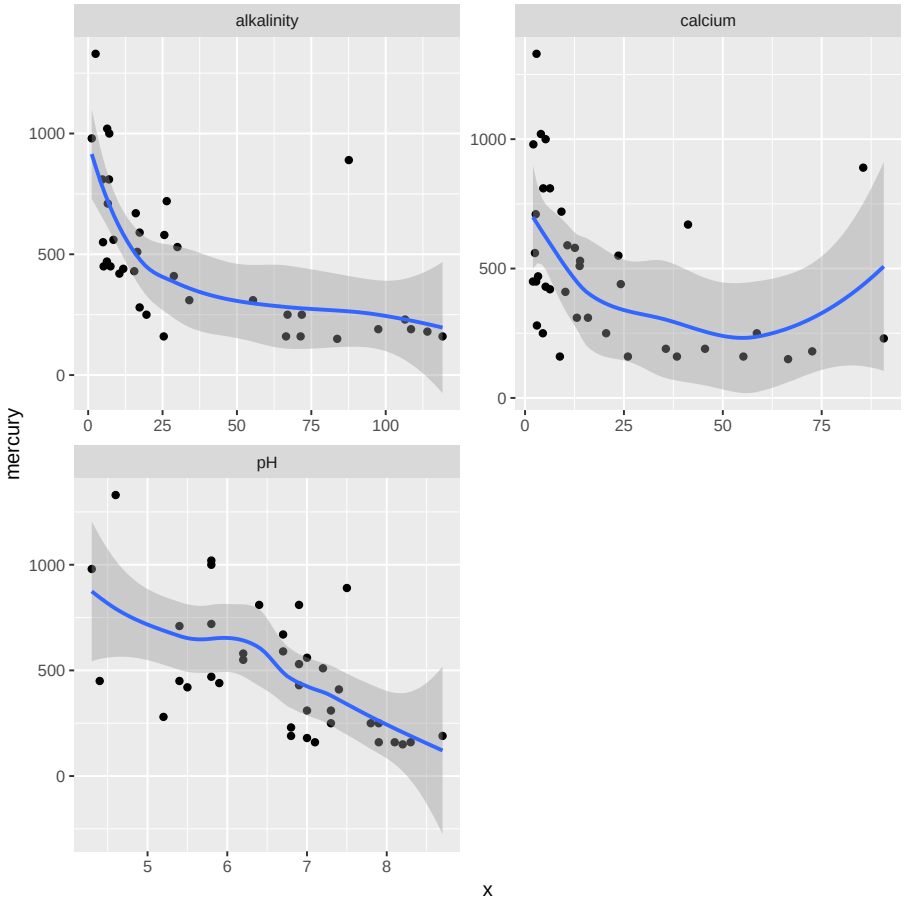

Figure 7: Scatter plots for fish mercury data

```r
mercury.1=lm(mercury~alkalinity+calcium+pH,data=mercury)
summary(mercury.1)
```

```
##
## Call:
## lm(formula = mercury ~ alkalinity + calcium + pH, data = mercury)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -371.46 -140.30   -3.97  106.31  551.98
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1221.451    279.406   4.372  0.00011 ***
## alkalinity    -4.681      2.014  -2.324  0.02622 *
## calcium        3.495      2.594   1.347  0.18685
## pH           -96.058     46.504  -2.066  0.04656 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 226.4 on 34 degrees of freedom
## Multiple R-squared:  0.4571,Adjusted R-squared:  0.4092
## F-statistic: 9.544 on 3 and 34 DF,  p-value: 0.0001024
```

Figure 8: Regression 1 for fish mercury data

```
mercury.2=lm(log(mercury)~log(alkalinity)+log(calcium)+pH,data=mercury)
summary(mercury.2)
```

```
##
## Call:
## lm(formula = log(mercury) ~ log(alkalinity) + log(calcium) +
##     pH, data = mercury)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.75244 -0.30191 -0.00783  0.23852  1.22932
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.55983    0.48981  15.434  < 2e-16 ***
## log(alkalinity) -0.45880    0.11807  -3.886 0.000449 ***
## log(calcium)     0.14702    0.10315   1.425 0.163185
## pH              -0.07998    0.10248  -0.780 0.440527
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4099 on 34 degrees of freedom
## Multiple R-squared:  0.6069,Adjusted R-squared:  0.5723
## F-statistic:  17.5 on 3 and 34 DF,  p-value: 4.808e-07
```

Figure 9: Regression 2 for fish mercury data

```
mercury.3=update(mercury.2,.~.-log(calcium)-pH)
```

Figure 10: Regression 3 for fish mercury data

```r
summary(mercury)
```

```
##     mercury          alkalinity        calcium             pH
##  Min.   : 150.0   Min.   :  1.20   Min.   : 2.000   Min.   :4.300
##  1st Qu.: 250.0   1st Qu.:  7.20   1st Qu.: 4.525   1st Qu.:5.800
##  Median : 445.0   Median : 18.45   Median :11.650   Median :6.850
##  Mean   : 488.4   Mean   : 37.15   Mean   :22.361   Mean   :6.634
##  3rd Qu.: 650.0   3rd Qu.: 66.88   3rd Qu.:33.200   3rd Qu.:7.300
##  Max.   :1330.0   Max.   :119.10   Max.   :90.700   Max.   :8.700
```

```r
new=data.frame(alkalinity=c(7,67))
new
```

```
##   alkalinity
## 1          7
## 2         67
```

```r
p=predict(mercury.3,new,interval="c")
cbind(new,p)
```

```
##   alkalinity      fit      lwr      upr
## 1          7 6.435714 6.252683 6.618745
## 2         67 5.536953 5.345056 5.728849
```

Figure 11: Prediction for fish mercury data

```
leukemia=read.csv("leukemia.csv", header=T)
leukemia
```

```
##    ag   wbc live
## 1   +    75    1
## 2   +   260    1
## 3   +  1000    1
## 4   +   700    1
## 5   +  3500    0
## 6   + 10000    0
## 7   -   300    1
## 8   -   900    0
## 9   -  1900    0
## 10  -  3100    0
## 11  -  7900    0
## 12  +   230    1
## 13  +   600    0
## 14  +  1700    0
## 15  +   940    1
## 16  +  5200    0
## 17  + 10000    0
## 18  -   400    0
## 19  -   530    0
## 20  -  2700    0
## 21  -  2600    0
## 22  - 10000    0
## 23  +   430    1
## 24  +  1050    1
## 25  +   540    0
## 26  +  3200    0
## 27  + 10000    1
## 28  -   440    1
## 29  -   150    0
## 30  -  1000    0
## 31  -  2800    0
## 32  -  2100    0
## 33  - 10000    0
```

Figure 12: Leukemia data

```
ggplot(leukemia,aes(x=factor(live),y=wbc))+geom_boxplot()
```

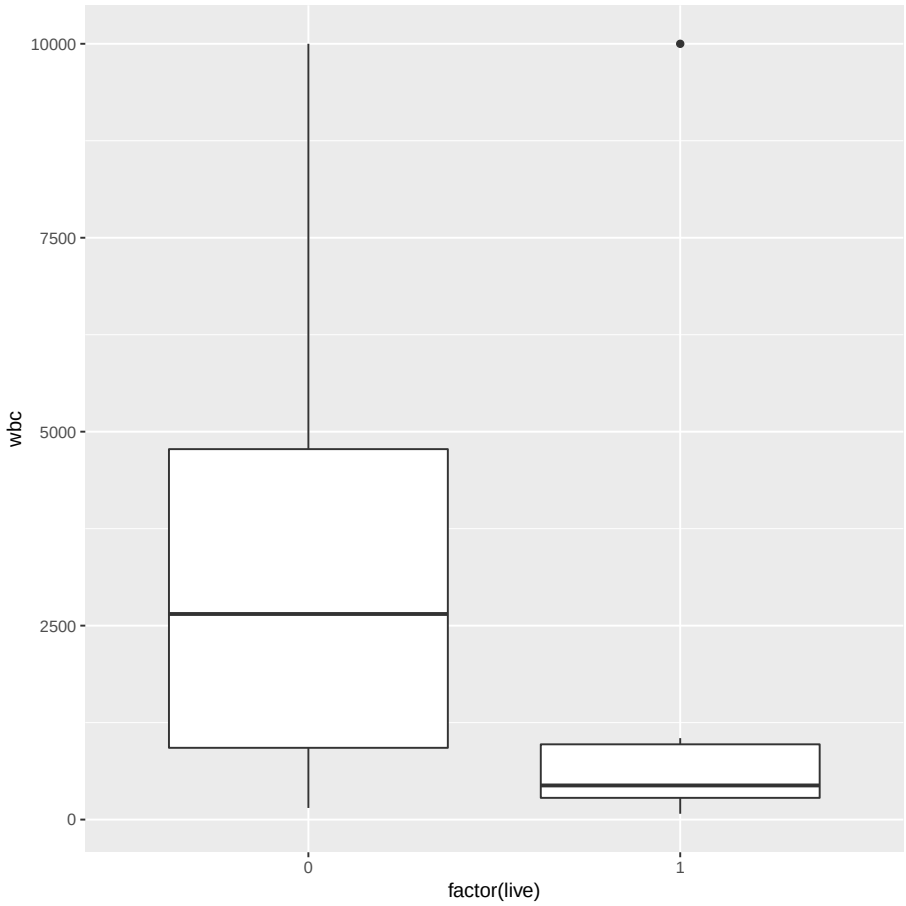

Figure 13: Boxplot of white blood cell count vs. survival

```r
leukemia.1=glm(live~log(wbc)+ag,family="binomial",data=leukemia)
summary(leukemia.1)
```

```
##
## Call:
## glm(formula = live ~ log(wbc) + ag, family = "binomial", data = leukemia)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6599  -0.6568  -0.2803   0.5286   2.1258
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   5.5433     3.0224   1.834   0.0666 .
## log(wbc)     -1.1088     0.4609  -2.405   0.0162 *
## ag+           2.5196     1.0907   2.310   0.0209 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 42.010  on 32  degrees of freedom
## Residual deviance: 26.833  on 30  degrees of freedom
## AIC: 32.833
##
## Number of Fisher Scoring iterations: 5
```

Figure 14: Logistic regression for leukemia data

```r
wbcs=c(500,1000,3000)
ags=c("+","-")
new=expand.grid(wbc=wbcs,ag=ags)
p=predict(leukemia.1,new,type="response")
cbind(new,p)
```

```
##    wbc ag          p
## 1  500  + 0.76358018
## 2 1000  + 0.59961853
## 3 3000  + 0.30699190
## 4  500  - 0.20633616
## 5 1000  - 0.10758150
## 6 3000  - 0.03443025
```

Figure 15: Predictions

```
painrelief0=read.table("drugcomp.txt",header=T)
lev=levels(painrelief0$rating)
lev
```

```
## [1] "fair"     "good"     "poor"     "verygood"
```

```
painrelief0 %>%
  mutate(rating=ordered(rating,lev[c(3,1,2,4)])) -> painrelief
painrelief
```

```
##    drug   rating frequency
## 1   c15     poor        17
## 2   c15     fair        18
## 3   c15     good        20
## 4   c15 verygood         5
## 5   c60     poor        30
## 6   c60     fair        25
## 7   c60     good        30
## 8   c60 verygood         8
## 9  z100     poor        10
## 10 z100     fair         4
## 11 z100     good        13
## 12 z100 verygood        34
```

```
painrelief$rating
```

```
##  [1] poor     fair     good     verygood poor     fair     good
##  [8] verygood poor     fair     good     verygood
## Levels: poor < fair < good < verygood
```

Figure 16: Pain relief data

```r
painrelief.1=polr(rating~drug, weight=frequency, data=painrelief)
drop1(painrelief.1,test="Chisq")
```

```
## Single term deletions
##
## Model:
## rating ~ drug
##        Df    AIC   LRT Pr(>Chi)
## <none>    557.96
## drug    2 595.87 41.91 7.93e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
new=data.frame(drug=levels(painrelief$drug))
p=predict(painrelief.1,new,type="probs")
cbind(new,p)
```

```
##   drug       poor       fair      good  verygood
## 1  c15 0.32446172 0.25034061 0.2972508 0.1279469
## 2  c60 0.34244375 0.25200600 0.2863636 0.1191867
## 3 z100 0.06486819 0.09848111 0.3327126 0.5039381
```

Figure 17: Model-fitting and predictions for pain relief data

```
clematis=read.csv("muenchow.csv",header=T)
str(clematis)

## 'data.frame': 96 obs. of  3 variables:
##  $ gender  : Factor w/ 2 levels "female","male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ wait    : int  1 1 2 2 4 4 5 5 6 6 ...
##  $ observed: Factor w/ 2 levels "no","yes": 2 2 2 2 2 2 2 2 2 2 ...

clematis %>% sample_n(20)

##    gender wait observed
## 1    male    7      yes
## 2  female   15      yes
## 3    male   68      yes
## 4  female    7      yes
## 5  female   29      yes
## 6  female   23      yes
## 7  female   30      yes
## 8  female   18      yes
## 9    male   19      yes
## 10   male    1      yes
## 11   male   61      yes
## 12 female   35      yes
## 13 female   29      yes
## 14 female   39      yes
## 15 female   19      yes
## 16 female   28      yes
## 17 female   90       no
## 18 female   75       no
## 19 female    2      yes
## 20   male   83      yes
```

Figure 18: Clematis data (structure and a few randomly-chosen rows)

```
ggplot(clematis,aes(x=gender,y=wait))+geom_boxplot()
```

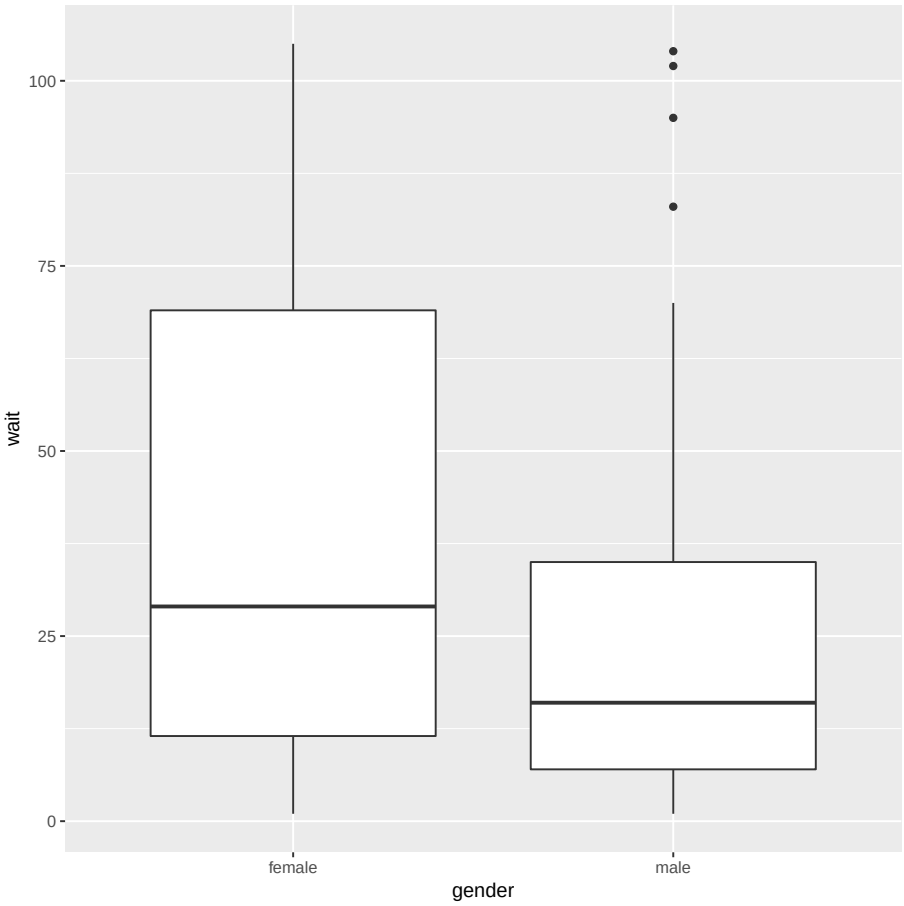

Figure 19: Boxplots of waiting times

```
y=with(clematis,Surv(wait,observed=="yes"))
y
```

```
##  [1]   1    1    2    2    4    4    5    5    6    6    6    7    7    8
## [15]   8    8    9    9    9   11   11   14   14   14   16   16   17   17
## [29]  18   19   19   19   27   27   30   31   35   36   40   43   54   61
## [43]  68   69   70   83   95  102+ 104+   1    2    4    4    5    6    7
## [57]   7    8    8    8    9   14   15   18   18   19   23   23   26   28
## [71]  29   29   29   30   32   35   35   37   39   43   56   57   59   67
## [85]  71   75   75+  78   81   90+  94+  96   96+ 100+ 102+ 105+
```

Figure 20: Construction of response variable

```
clematis.1=coxph(y~gender,data=clematis)
summary(clematis.1)
```

```
## Call:
## coxph(formula = y ~ gender, data = clematis)
##
##   n= 96, number of events= 87
##
##              coef exp(coef) se(coef)     z Pr(>|z|)
## gendermale 0.5069    1.6602   0.2170 2.336   0.0195 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##            exp(coef) exp(-coef) lower .95 upper .95
## gendermale      1.66     0.6023     1.085      2.54
##
## Concordance= 0.571  (se = 0.029 )
## Likelihood ratio test= 5.47  on 1 df,   p=0.02
## Wald test            = 5.46  on 1 df,   p=0.02
## Score (logrank) test = 5.57  on 1 df,   p=0.02
```

Figure 21: Cox model 1

```
clematis.0=coxph(y~1,data=clematis)
anova(clematis.0,clematis.1)

## Analysis of Deviance Table
##  Cox model: response is  y
##  Model 1: ~ 1
##  Model 2: ~ gender
##    loglik  Chisq Df P(>|Chi|)
## 1 -331.66
## 2 -328.93 5.4726  1   0.01932 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 22: Cox model 2 and comparison

```
essay=read.csv("essay.csv",header=T)
essay

##    ability   method score
## 1     none bluebook    23
## 2     none bluebook    32
## 3     none bluebook    25
## 4     some bluebook    29
## 5     some bluebook    30
## 6     some bluebook    34
## 7     lots bluebook    31
## 8     lots bluebook    36
## 9     lots bluebook    33
## 10    none computer    32
## 11    none computer    26
## 12    none computer    26
## 13    some computer    34
## 14    some computer    41
## 15    some computer    35
## 16    lots computer    23
## 17    lots computer    26
## 18    lots computer    32
```

Figure 23: Essay marks data

```
essay %>% mutate(ability.ord=ordered(ability,c("none","some","lots"))) %>%
  group_by(ability.ord,method) %>%
  summarize(mean.score=mean(score)) -> essay.means
essay.means
```

```
## # A tibble: 6 x 3
## # Groups:   ability.ord [3]
##   ability.ord method   mean.score
##   <ord>       <fct>         <dbl>
## 1 none        bluebook       26.7
## 2 none        computer       28
## 3 some        bluebook       31
## 4 some        computer       36.7
## 5 lots        bluebook       33.3
## 6 lots        computer       27
```

Figure 24: Essay marks means

```
essay.1=aov(score~ability+method,data=essay)
summary(essay.1)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## ability      2 127.44   63.72   3.223 0.0706 .
## method       1   0.22    0.22   0.011 0.9171
## Residuals   14 276.78   19.77
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 25: Essay marks analysis 1

```
essay.2=aov(score~ability*method,data=essay)
summary(essay.2)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## ability         2 127.44   63.72   4.606 0.0328 *
## method          1   0.22    0.22   0.016 0.9012
## ability:method  2 110.78   55.39   4.004 0.0465 *
## Residuals      12 166.00   13.83
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 26: Essay marks analysis 2

```r
essay %>% filter(method=="computer") %>%
  aov(score~ability,data=.) -> essay.3
summary(essay.3)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## ability      2 169.56   84.78   5.373  0.046 *
## Residuals    6  94.67   15.78
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
TukeyHSD(essay.3,conf.level=0.90)
```

```
##   Tukey multiple comparisons of means
##     90% family-wise confidence level
##
## Fit: aov(formula = score ~ ability, data = .)
##
## $ability
##                diff        lwr      upr     p adj
## none-lots 1.000000 -7.1604402  9.16044 0.9493774
## some-lots 9.666667  1.5062264 17.82711 0.0557235
## some-none 8.666667  0.5062264 16.82711 0.0820210
```

Figure 27: Essay marks analysis 3

```
essay %>% filter(method=="bluebook") %>%
  aov(score~ability,data=.) -> essay.4
summary(essay.4)

##             Df Sum Sq Mean Sq F value Pr(>F)
## ability      2  68.67   34.33   2.888  0.132
## Residuals    6  71.33   11.89

TukeyHSD(essay.4,conf.level=0.90)

##   Tukey multiple comparisons of means
##     90% family-wise confidence level
##
## Fit: aov(formula = score ~ ability, data = .)
##
## $ability
##                 diff        lwr       upr     p adj
## none-lots -6.666667 -13.750385  0.417052 0.1207974
## some-lots -2.333333  -9.417052  4.750385 0.7004165
## some-none  4.333333  -2.750385 11.417052 0.3395194
```

Figure 28: Essay marks analysis 4

```
pottery=read.table("pottery.txt",header=T)
pottery
```

```
##      Al   Fe   Mg   Ca   Na         Site
## 1  14.4 7.00 4.30 0.15 0.51   Llanederyn
## 2  13.8 7.08 3.43 0.12 0.17   Llanederyn
## 3  14.6 7.09 3.88 0.13 0.20   Llanederyn
## 4  11.5 6.37 5.64 0.16 0.14   Llanederyn
## 5  13.8 7.06 5.34 0.20 0.20   Llanederyn
## 6  10.9 6.26 3.47 0.17 0.22   Llanederyn
## 7  10.1 4.26 4.26 0.20 0.18   Llanederyn
## 8  11.6 5.78 5.91 0.18 0.16   Llanederyn
## 9  11.1 5.49 4.52 0.29 0.30   Llanederyn
## 10 13.4 6.92 7.23 0.28 0.20   Llanederyn
## 11 12.4 6.13 5.69 0.22 0.54   Llanederyn
## 12 13.1 6.64 5.51 0.31 0.24   Llanederyn
## 13 12.7 6.69 4.45 0.20 0.22   Llanederyn
## 14 12.5 6.44 3.94 0.22 0.23   Llanederyn
## 15 11.8 5.44 3.94 0.30 0.04     Caldicot
## 16 11.6 5.39 3.77 0.29 0.06     Caldicot
## 17 18.3 1.28 0.67 0.03 0.03 IslandThorns
## 18 15.8 2.39 0.63 0.01 0.04 IslandThorns
## 19 18.0 1.50 0.67 0.01 0.06 IslandThorns
## 20 18.0 1.88 0.68 0.01 0.04 IslandThorns
## 21 20.8 1.51 0.72 0.07 0.10 IslandThorns
## 22 17.7 1.12 0.56 0.06 0.06  AshleyRails
## 23 18.3 1.14 0.67 0.06 0.05  AshleyRails
## 24 16.7 0.92 0.53 0.01 0.05  AshleyRails
## 25 14.8 2.74 0.67 0.03 0.05  AshleyRails
## 26 19.1 1.64 0.60 0.10 0.03  AshleyRails
```

Figure 29: Pottery data

```
ggplot(pottery,aes(x=Site,y=Fe))+geom_boxplot()
```

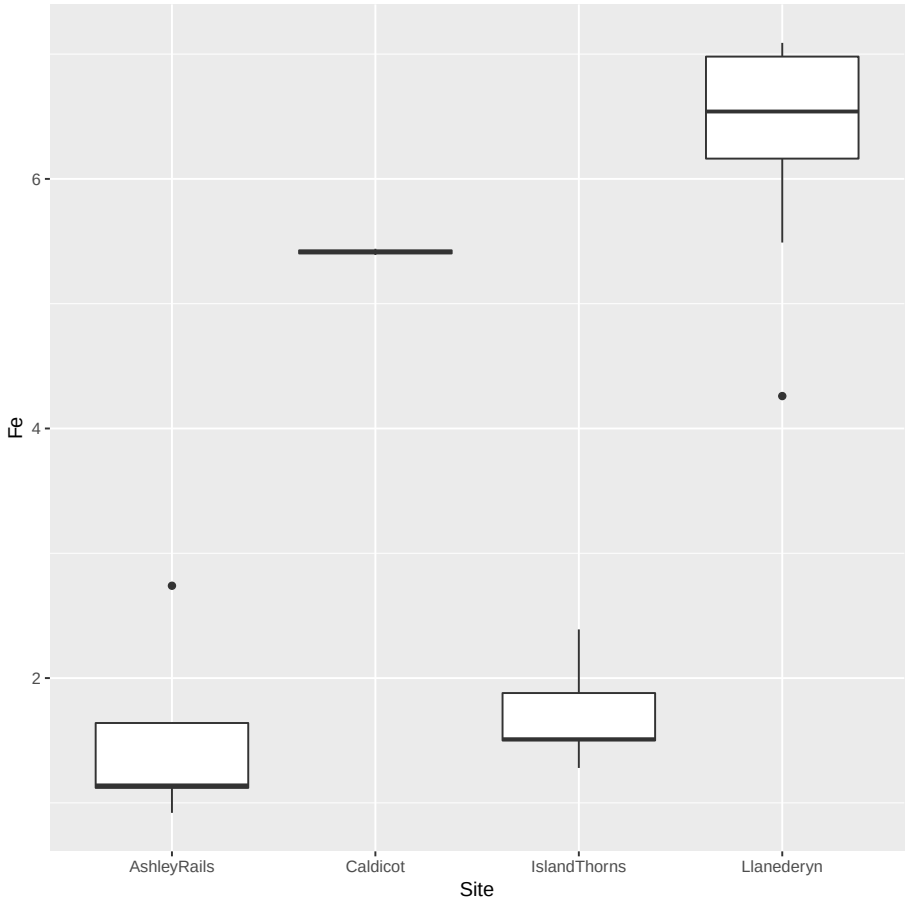

Figure 30: Boxplots of pottery data

```
levels(pottery$Site)

## [1] "AshleyRails"  "Caldicot"     "IslandThorns" "Llanederyn"

c.a=c(1,0,-1,0)
c.b=c(-1/2,1/2,-1/2,1/2)
c.c=c(0,1,0,-1)
m=cbind(c.a,c.b,c.c)
contrasts(pottery$Site)=m
```

Figure 31: Computations for pottery data

```r
pottery.1=lm(Fe~Site,data=pottery)
summary(pottery.1)
```

```
##
## Call:
## lm(formula = Fe ~ Site, data = pottery)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.11214 -0.33954  0.01143  0.49036  1.22800
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7528     0.1738  21.587 2.68e-16 ***
## Sitec.a      -0.1000     0.2231  -0.448   0.6584
## Sitec.b       4.2816     0.3477  12.314 2.42e-11 ***
## Sitec.c      -0.4786     0.2667  -1.795   0.0865 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7055 on 22 degrees of freedom
## Multiple R-squared:  0.9246,Adjusted R-squared:  0.9143
## F-statistic: 89.88 on 3 and 22 DF,  p-value: 1.679e-12
```

Figure 32: Analysis for pottery data

```
ggplot(painrelief,aes(x=drug,weight=frequency,fill=rating))+
  geom_bar(position="dodge")
```

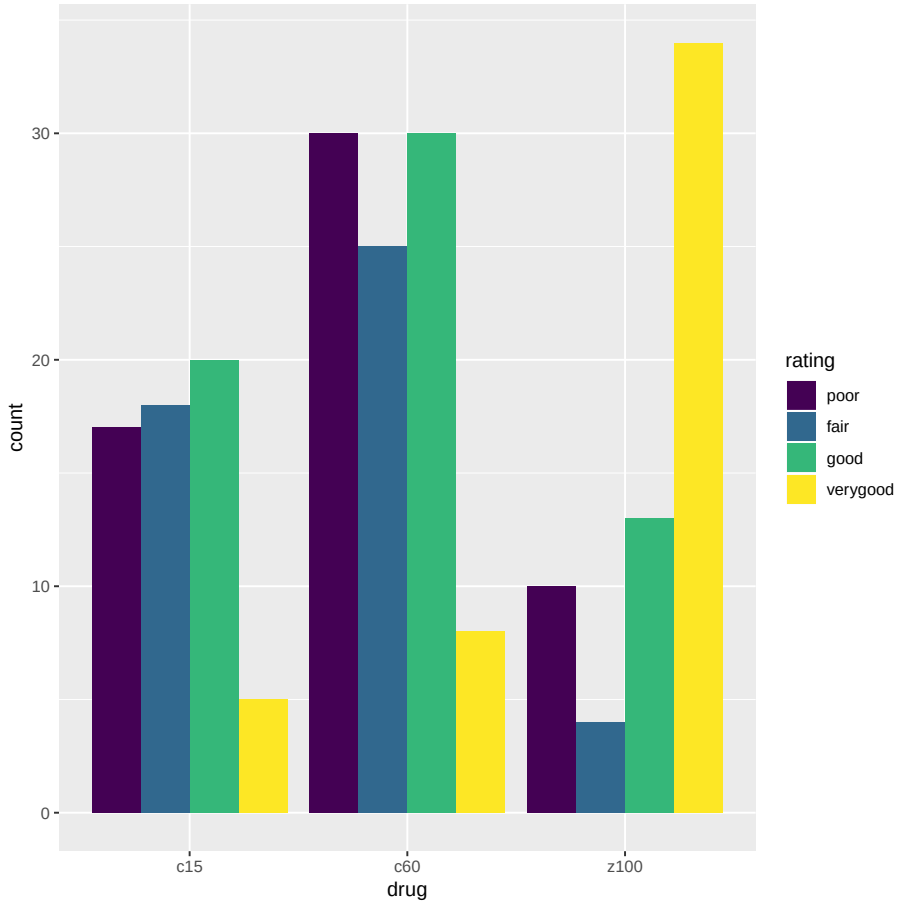

Figure 33: Grouped bar chart of pain relief data

```r
genders=with(clematis,levels(gender))
new=data.frame(gender=genders)
p=survfit(clematis.1,new)
new
```

```
##   gender
## 1 female
## 2   male
```

```r
ggsurvplot(p)
```

```
## Error in .get_data(fit, data = data, complain = FALSE):
The `data` argument should be provided either to ggsurvfit or
survfit.
```

Figure 34: Predictions and survival plot

```
ggplot(essay.means,aes(x=ability.ord,y=mean.score,
    colour=method,group=method))+
  geom_point()+geom_line()
```

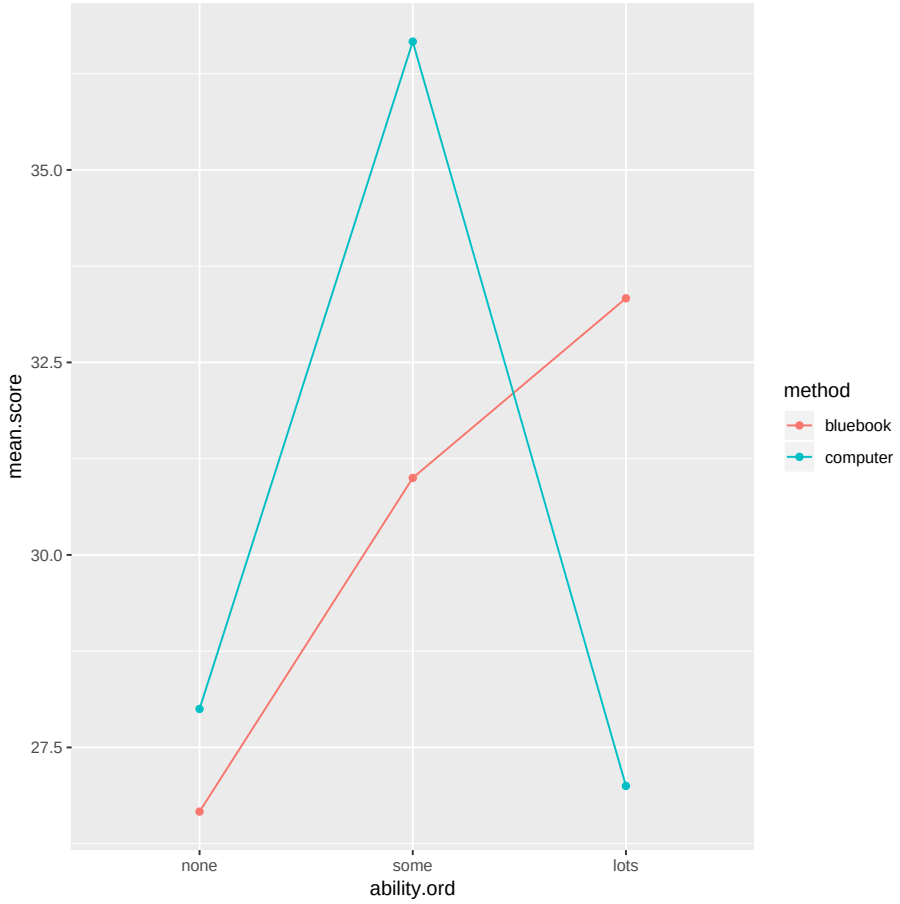

Figure 35: Essay marks plot