In [1]:

```
library(caretEnsemble)
library(RColorBrewer)
library(tm)
library(datarium)
library(leaps)
library(glmnet)
library(pls)
library(gam)
library(splines)
library(MVA)
library(nortest)
library(mvnormtest)
library(pastecs)
library(mvtnorm)
library(igraph)
library(dplyr)
library(ggplot2)
library(ggraph)
library(caret)
library(car)
library(mlbench)
library(tidyverse)
library(MASS)
library(ISLR)
library(psych)
library(faraway)
library(pls)
library(Matrix)
library(stats)
library(biotools)
library(ggpubr)
library(broom)
library(leaps)
library(tidyverse)
library(funModeling)
library(Hmisc)
library(rpart)
library(readr)
library(party)
library(partykit)
library(rpart.plot)
library(stringr)
library(reshape2)
library(pROC)
library(corrplot)
library(InformationValue)
library(foreign)
library(nnet)
#install.packages('reshape')
library(reshape)
```

```
Loading required package: NLP

Loading required package: Matrix

Loaded glmnet 4.1-2
```

```
Attaching package: ‘pls’

The following object is masked from ‘package:stats’:

    loadings

Loading required package: splines

Loading required package: foreach

Loaded gam 1.20

Loading required package: HSAUR2

Loading required package: tools

Attaching package: ‘igraph’

The following objects are masked from ‘package:stats’:

    decompose, spectrum

The following object is masked from ‘package:base’:

    union


Attaching package: ‘dplyr’

The following objects are masked from ‘package:igraph’:

    as_data_frame, groups, union

The following objects are masked from ‘package:pastecs’:

    first, last

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union


Attaching package: ‘ggplot2’
```

```
The following object is masked from ‘package:NLP’:

    annotate


The following object is masked from ‘package:caretEnsemble’:

    autoplot


Loading required package: lattice


Attaching package: ‘caret’


The following object is masked from ‘package:pls’:

    R2


Loading required package: carData


Attaching package: ‘car’


The following object is masked from ‘package:dplyr’:

    recode


── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

✓ tibble  3.1.3     ✓ purrr   0.3.4
✓ tidyr   1.1.3     ✓ stringr 1.4.0
✓ readr   2.0.1     ✓ forcats 0.5.1

── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✕ purrr::accumulate()     masks foreach::accumulate()
✕ ggplot2::annotate()     masks NLP::annotate()
✕ tibble::as_data_frame() masks dplyr::as_data_frame(), igraph::as_data_frame()
✕ ggplot2::autoplot()     masks caretEnsemble::autoplot()
✕ purrr::compose()        masks igraph::compose()
✕ tidyr::crossing()       masks igraph::crossing()
✕ tidyr::expand()         masks Matrix::expand()
✕ tidyr::extract()        masks pastecs::extract()
✕ dplyr::filter()         masks stats::filter()
✕ dplyr::first()          masks pastecs::first()
✕ dplyr::groups()         masks igraph::groups()
✕ dplyr::lag()            masks stats::lag()
✕ dplyr::last()           masks pastecs::last()
✕ purrr::lift()           masks caret::lift()
✕ tidyr::pack()           masks Matrix::pack()
✕ car::recode()           masks dplyr::recode()
✕ purrr::simplify()       masks igraph::simplify()
✕ purrr::some()           masks car::some()
✕ tidyr::unpack()         masks Matrix::unpack()
✕ purrr::when()           masks foreach::when()
```

```
Attaching package: ‘MASS’

The following object is masked from ‘package:dplyr’:

    select


Attaching package: ‘psych’

The following object is masked from ‘package:car’:

    logit

The following objects are masked from ‘package:ggplot2’:

    %+%, alpha


Attaching package: ‘faraway’

The following object is masked from ‘package:psych’:

    logit

The following objects are masked from ‘package:car’:

    logit, vif

The following object is masked from ‘package:lattice’:

    melanoma

The following objects are masked from ‘package:HSAUR2’:

    epilepsy, toenail

---
biotools version 4.2

Loading required package: Hmisc

Loading required package: survival

Attaching package: ‘survival’

The following objects are masked from ‘package:faraway’:

    rats, solder
```

```
The following object is masked from ‘package:caret’:

    cluster


Loading required package: Formula


Attaching package: ‘Hmisc’


The following object is masked from ‘package:psych’:

    describe


The following objects are masked from ‘package:dplyr’:

    src, summarize


The following objects are masked from ‘package:base’:

    format.pval, units


funModeling v.1.9.4 :)
Examples and tutorials at livebook.datascienceheroes.com
 / Now in Spanish: librovivodecienciadedatos.ai


Attaching package: ‘rpart’


The following object is masked from ‘package:faraway’:

    solder


Loading required package: grid

Loading required package: modeltools

Loading required package: stats4


Attaching package: ‘modeltools’


The following object is masked from ‘package:car’:

    Predict


The following object is masked from ‘package:igraph’:

    clusters


Loading required package: strucchange
```

```
Loading required package: zoo

Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric

Loading required package: sandwich

Attaching package: ‘strucchange’

The following object is masked from ‘package:stringr’:

    boundary

Loading required package: libcoin

Attaching package: ‘partykit’

The following objects are masked from ‘package:party’:

    cforest, ctree, ctree_control, edge_simple, mob, mob_control,
    node_barplot, node_bivplot, node_boxplot, node_inner, node_surv,
    node_terminal, varimp


Attaching package: ‘reshape2’

The following object is masked from ‘package:tidyr’:

    smiths

Type 'citation("pROC")' for a citation.

Attaching package: ‘pROC’

The following objects are masked from ‘package:stats’:

    cov, smooth, var

corrplot 0.90 loaded

Attaching package: ‘corrplot’
```

```
The following object is masked from ‘package:pls’:

    corrplot


Attaching package: ‘InformationValue’

The following objects are masked from ‘package:caret’:

    confusionMatrix, precision, sensitivity, specificity


Attaching package: ‘reshape’

The following objects are masked from ‘package:reshape2’:

    colsplit, melt, recast

The following objects are masked from ‘package:tidyr’:

    expand, smiths

The following object is masked from ‘package:dplyr’:

    rename

The following object is masked from ‘package:Matrix’:

    expand
```

In [2]:

```
data01 <- read.dta("https://stats.idre.ucla.edu/stat/data/hsbdemo.dta")
head(data01)
```

A data.frame: 6 × 13

	id	female	ses	schtyp	prog	read	write	math	science	socst	honors	awards	c
	<dbl>	<fct>	<fct>	<fct>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fct>	<dbl>	<int
1	45	female	low	public	vocation	34	35	41	29	26	not enrolled	0	
2	108	male	middle	public	general	34	33	41	36	36	not enrolled	0	
3	15	male	high	public	vocation	39	39	44	26	42	not enrolled	0	
4	67	male	low	public	vocation	37	37	42	33	32	not enrolled	0	

	id	female	ses	schtyp	prog	read	write	math	science	socst	honors	awards	c
	<dbl>	<fct>	<fct>	<fct>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fct>	<dbl>	<int
5	153	male	middle	public	vocation	39	31	40	39	51	not enrolled	0	
6	51	female	high	public	general	42	36	42	31	39	not enrolled	0	

(a) Make a table showing the proportion of males and females choosing the three different programs. Comment on the difference. Repeat this comparison but for SES rather than gender.

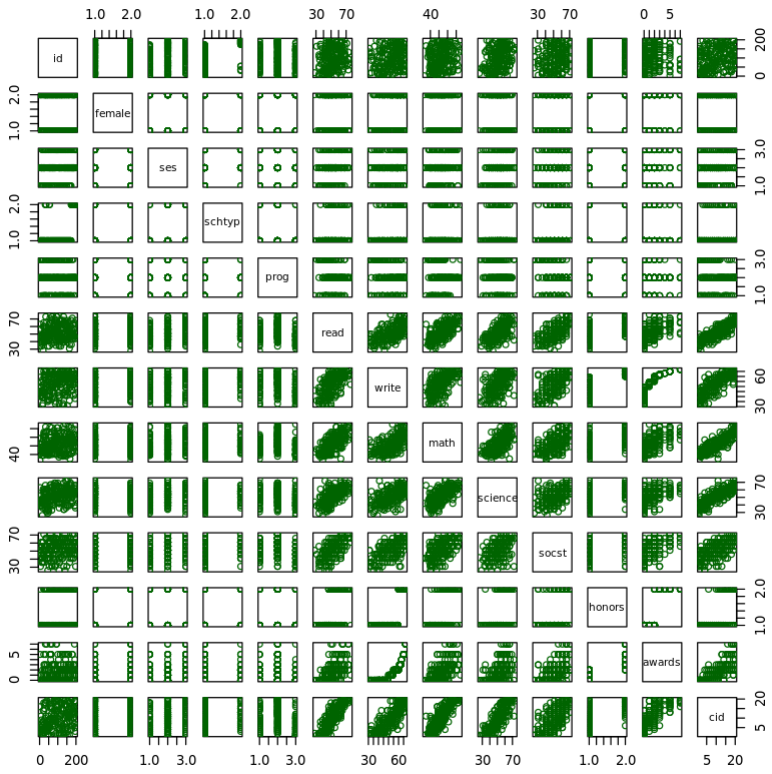
In [3]:
```
sum(is.na(data01))
```

0

In [4]:
```
dim(data01)
```

200 · 13

In [5]:
```
pairs(data01, col = 'darkgreen')
```



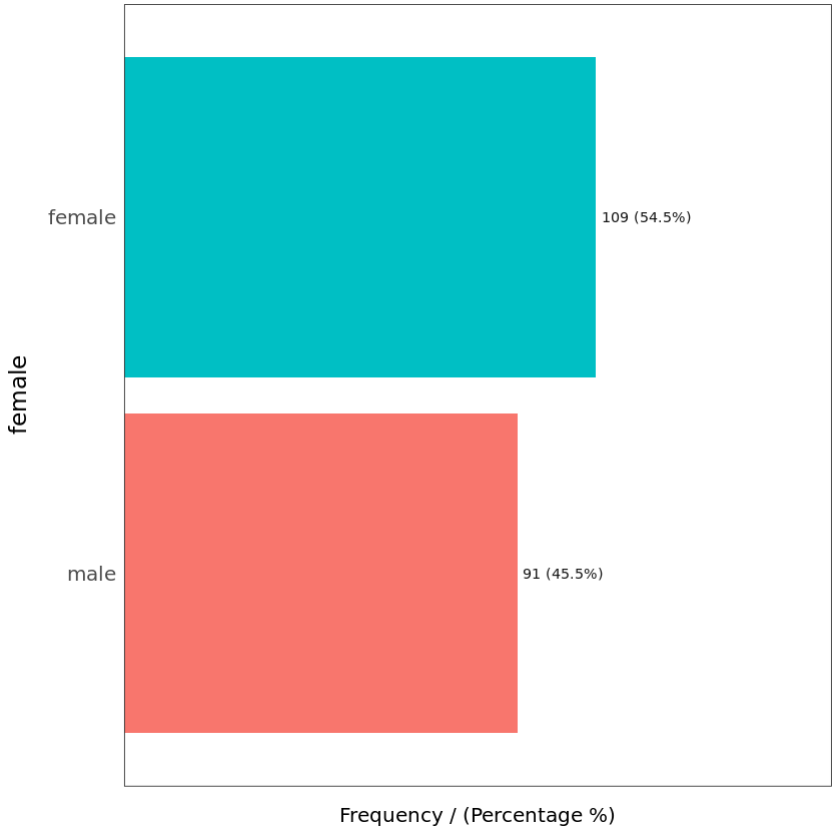
In [6]:
```
glimpse(data01)
print(status(data01))
freq(data01)
print(profiling_num(data01))
plot_num(data01)
describe(data01)
```

```
Rows: 200
Columns: 13
$ id      <dbl> 45, 108, 15, 67, 153, 51, 164, 133, 2, 53, 1, 128, 16, 106, 89…
$ female  <fct> female, male, male, male, male, female, male, male, female, ma…
$ ses     <fct> low, middle, high, low, middle, high, middle, middle, middle, …
$ schtyp  <fct> public, public, public, public, public, public, public, public…
$ prog    <fct> vocation, general, vocation, vocation, vocation, general, voca…
$ read    <dbl> 34, 34, 39, 37, 39, 42, 31, 50, 39, 34, 34, 39, 47, 36, 35, 44…
$ write   <dbl> 35, 33, 39, 37, 31, 36, 36, 31, 41, 37, 44, 33, 31, 44, 35, 44…
$ math    <dbl> 41, 41, 44, 42, 40, 42, 46, 40, 33, 46, 40, 38, 44, 37, 40, 39…
$ science <dbl> 29, 36, 26, 33, 39, 31, 39, 34, 42, 39, 39, 47, 36, 42, 51, 34…
$ socst   <dbl> 26, 36, 42, 32, 51, 39, 46, 31, 41, 31, 41, 41, 36, 41, 33, 46…
$ honors  <fct> not enrolled, not enrolled, not enrolled, not enrolled, not en…
$ awards  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…
$ cid     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,…
        variable q_zeros p_zeros q_na p_na q_inf p_inf    type unique
id            id       0    0.00    0    0     0     0 numeric    200
female    female       0    0.00    0    0     0     0  factor      2
ses          ses       0    0.00    0    0     0     0  factor      3
schtyp    schtyp       0    0.00    0    0     0     0  factor      2
prog        prog       0    0.00    0    0     0     0  factor      3
read        read       0    0.00    0    0     0     0 numeric     30
write      write       0    0.00    0    0     0     0 numeric     29
math        math       0    0.00    0    0     0     0 numeric     40
science  science       0    0.00    0    0     0     0 numeric     34
socst      socst       0    0.00    0    0     0     0 numeric     22
honors    honors       0    0.00    0    0     0     0  factor      2
awards    awards      72    0.36    0    0     0     0 numeric      7
cid          cid       0    0.00    0    0     0     0 integer     20
```

```
Warning message:
“`guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = "none")` instead.”
  female frequency percentage cumulative_perc
1 female       109       54.5            54.5
2   male        91       45.5           100.0
```

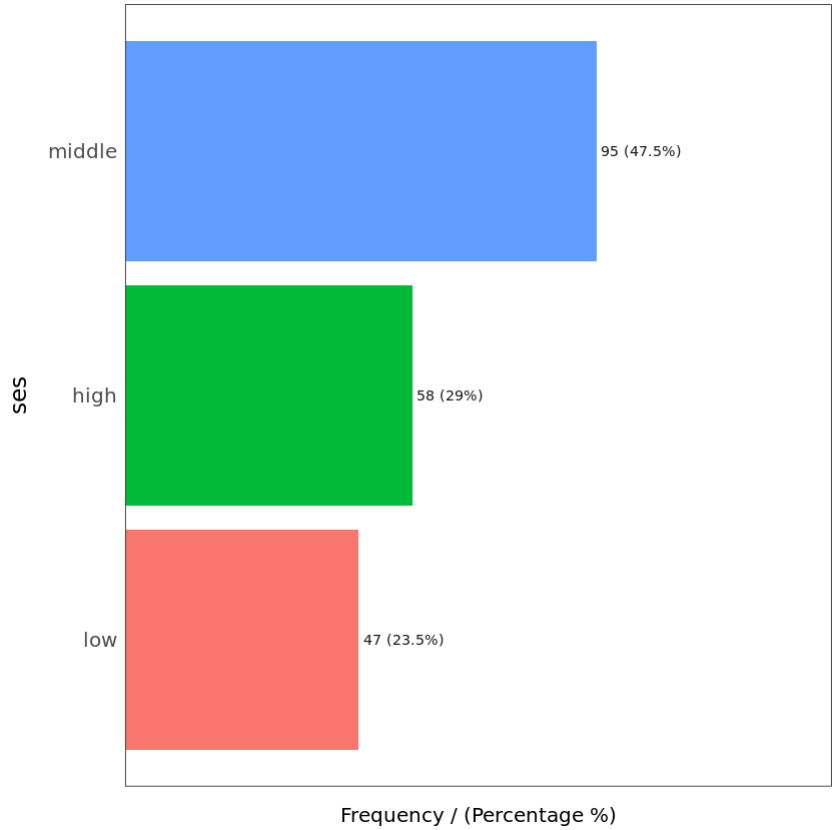
```
Warning message:
“`guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = "none")` instead.”
     ses frequency percentage cumulative_perc
1 middle        95       47.5            47.5
2   high        58       29.0            76.5
3    low        47       23.5           100.0
```

```
Warning message:
“`guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = "none")` instead.”
```




```
   schtyp frequency percentage cumulative_perc
1  public       168         84              84
2 private        32         16             100
```

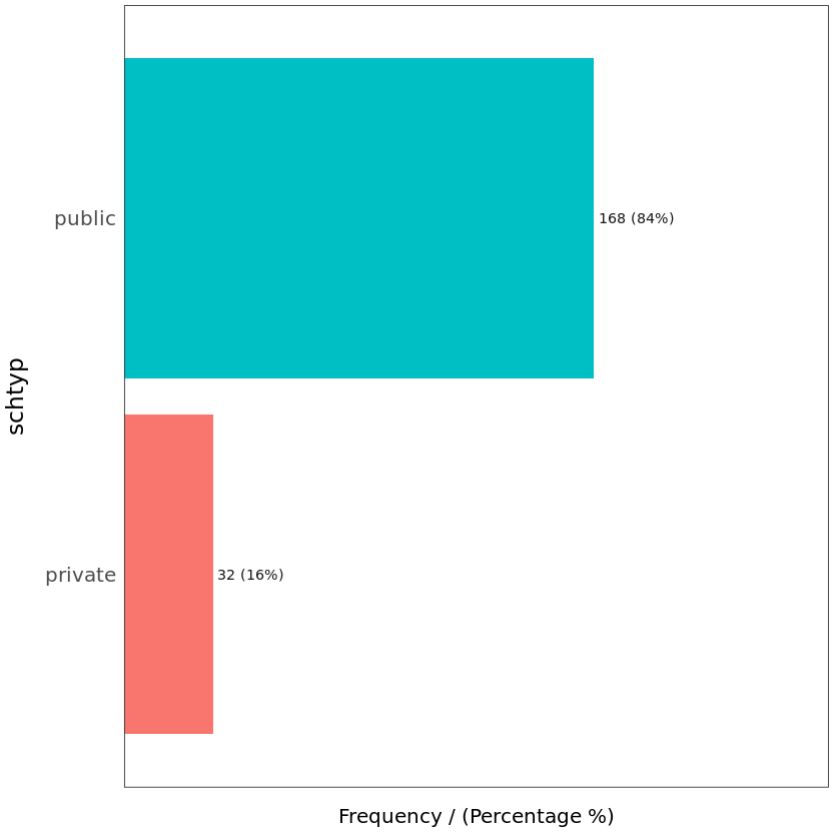
```
Warning message:
“`guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = "none")` instead.”
```

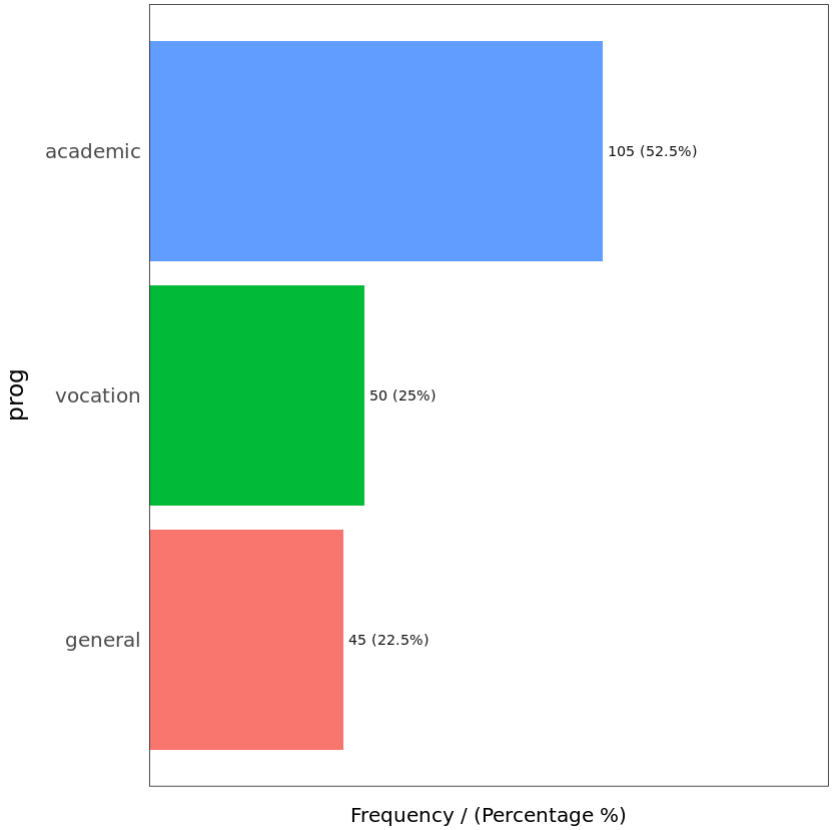



```
prog frequency percentage cumulative_perc
```

```
1 academic       105       52.5           52.5
2 vocation        50       25.0           77.5
3  general        45       22.5          100.0
```

```
Warning message:
“`guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = "none")` instead.”
```





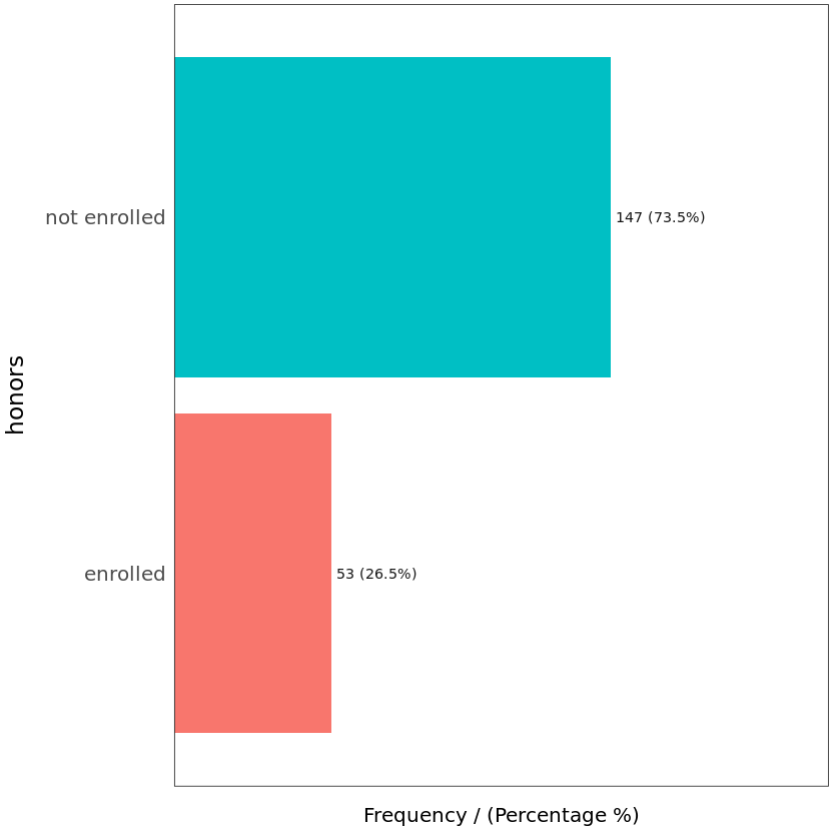

```
honors frequency percentage cumulative_perc
```

```
1 not enrolled       147       73.5           73.5
2     enrolled        53       26.5          100.0
```

'Variables processed: female, ses, schtyp, prog, honors'

```
  variable    mean   std_dev variation_coef  p_01  p_05  p_25  p_50   p_75
1       id 100.500 57.879185      0.5759123  2.99 10.95 50.75 100.5 150.25
2     read  52.230 10.252937      0.1963036 33.97 36.00 44.00  50.0  60.00
3    write  52.775  9.478586      0.1796037 31.00 35.95 45.75  54.0  60.00
4     math  52.645  9.368448      0.1779551 36.98 39.00 45.00  52.0  59.00
5  science  51.850  9.900891      0.1909526 30.98 34.00 44.00  53.0  58.00
6    socst  52.405 10.735793      0.2048620 26.00 31.00 46.00  52.0  61.00
7   awards   1.670  1.818691      1.0890367  0.00  0.00  0.00   1.0   2.00
8      cid  10.430  5.801152      0.5561987  1.00  1.00  5.00  10.5  15.00
    p_95   p_99    skewness kurtosis   iqr       range_98      range_80
1 190.05 198.01  0.00000000 1.799940 99.50 [2.99, 198.01] [20.9, 180.1]
2  68.00  73.03  0.19483729 2.363052 16.00 [33.97, 73.03]    [39, 66.2]
3  65.00  67.00 -0.47841577 2.238527 14.25       [31, 67]      [39, 65]
4  70.05  73.02  0.28441149 2.337319 14.00 [36.98, 73.02]    [40, 65.1]
5  66.05  72.00 -0.18722772 2.428308 14.00    [30.98, 72]    [39, 64.1]
6  66.00  71.00 -0.37866236 2.458539 15.00       [26, 71]      [36, 66]
7   5.00   7.00  1.17981930 3.864340  2.00         [0, 7]        [0, 5]
8  19.05  20.00  0.01530049 1.803955 10.00        [1, 20]       [2, 19]
```

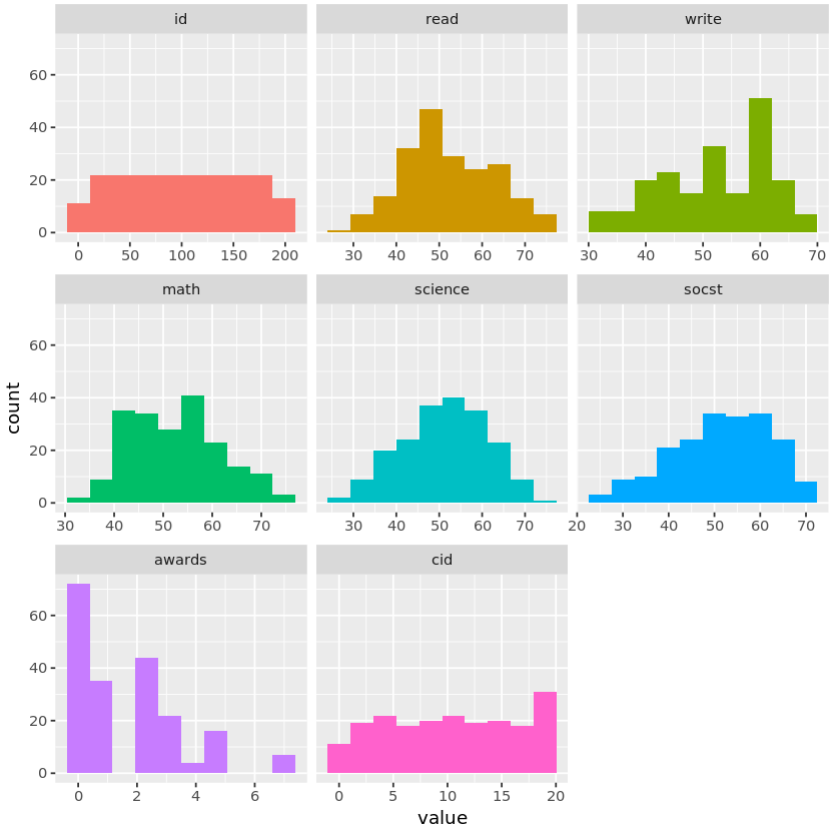
```
Warning message:
“`guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = "none")` instead.”
```




```
data01

 13  Variables      200  Observations
--------------------------------------------------------------------------------
id
       n  missing distinct     Info     Mean      Gmd      .05      .10
     200        0      200        1    100.5       67    10.95    20.90
     .25      .50      .75      .90      .95
   50.75   100.50   150.25   180.10   190.05
```

```
lowest :   1   2   3   4   5, highest: 196 197 198 199 200
---------------------------------------------------------------------------
female
       n  missing distinct
     200        0        2

Value        male female
Frequency      91    109
Proportion  0.455  0.545
---------------------------------------------------------------------------
ses
       n  missing distinct
     200        0        3

Value         low middle   high
Frequency      47     95     58
Proportion  0.235  0.475  0.290
---------------------------------------------------------------------------
schtyp
       n  missing distinct
     200        0        2

Value       public private
Frequency      168      32
Proportion    0.84    0.16
---------------------------------------------------------------------------
prog
       n  missing distinct
     200        0        3

Value       general academic vocation
Frequency        45      105       50
Proportion    0.225    0.525    0.250
---------------------------------------------------------------------------
read
       n  missing distinct     Info     Mean      Gmd      .05      .10
     200        0       30    0.994    52.23    11.69     36.0     39.0
     .25      .50      .75      .90      .95
    44.0     50.0     60.0     66.2     68.0

lowest : 28 31 34 35 36, highest: 66 68 71 73 76
---------------------------------------------------------------------------
write
       n  missing distinct     Info     Mean      Gmd      .05      .10
     200        0       29    0.995    52.77    10.77    35.95    39.00
     .25      .50      .75      .90      .95
   45.75    54.00    60.00    65.00    65.00

lowest : 31 33 35 36 37, highest: 61 62 63 65 67
---------------------------------------------------------------------------
math
       n  missing distinct     Info     Mean      Gmd      .05      .10
     200        0       40    0.999    52.65    10.72    39.00    40.00
     .25      .50      .75      .90      .95
   45.00    52.00    59.00    65.10    70.05

lowest : 33 35 37 38 39, highest: 70 71 72 73 75
---------------------------------------------------------------------------
science
       n  missing distinct     Info     Mean      Gmd      .05      .10
     200        0       34    0.995    51.85     11.3    34.00    39.00
```

```
     .25      .50      .75      .90      .95
   44.00    53.00    58.00    64.10    66.05

lowest : 26 29 31 33 34, highest: 66 67 69 72 74
---------------------------------------------------------------------------
socst
       n  missing distinct     Info     Mean      Gmd      .05      .10
     200        0       22    0.984    52.41    12.15       31       36
     .25      .50      .75      .90      .95
      46       52       61       66       66

lowest : 26 31 32 33 36, highest: 57 58 61 66 71
---------------------------------------------------------------------------
honors
       n  missing distinct
     200        0        2

Value      not enrolled     enrolled
Frequency           147           53
Proportion        0.735        0.265
---------------------------------------------------------------------------
awards
       n  missing distinct     Info     Mean      Gmd
     200        0        7    0.935     1.67     1.91

lowest : 0 1 2 3 4, highest: 2 3 4 5 7

Value          0     1     2     3     4     5     7
Frequency     72    35    44    22     4    16     7
Proportion 0.360 0.175 0.220 0.110 0.020 0.080 0.035
---------------------------------------------------------------------------
cid
       n  missing distinct     Info     Mean      Gmd      .05      .10
     200        0       20    0.997    10.43    6.706     1.00     2.00
     .25      .50      .75      .90      .95
    5.00    10.50    15.00    19.00    19.05

lowest :  1  2  3  4  5, highest: 16 17 18 19 20

Value          1     2     3     4     5     6     7     8     9    10    11
Frequency     11    10     9    11    11     9     9    11     9    10    12
Proportion 0.055 0.050 0.045 0.055 0.055 0.045 0.045 0.055 0.045 0.050 0.060

Value         12    13    14    15    16    17    18    19    20
Frequency     10     9    10    10    11     7    10    11    10
Proportion 0.050 0.045 0.050 0.050 0.055 0.035 0.050 0.055 0.050
---------------------------------------------------------------------------
```



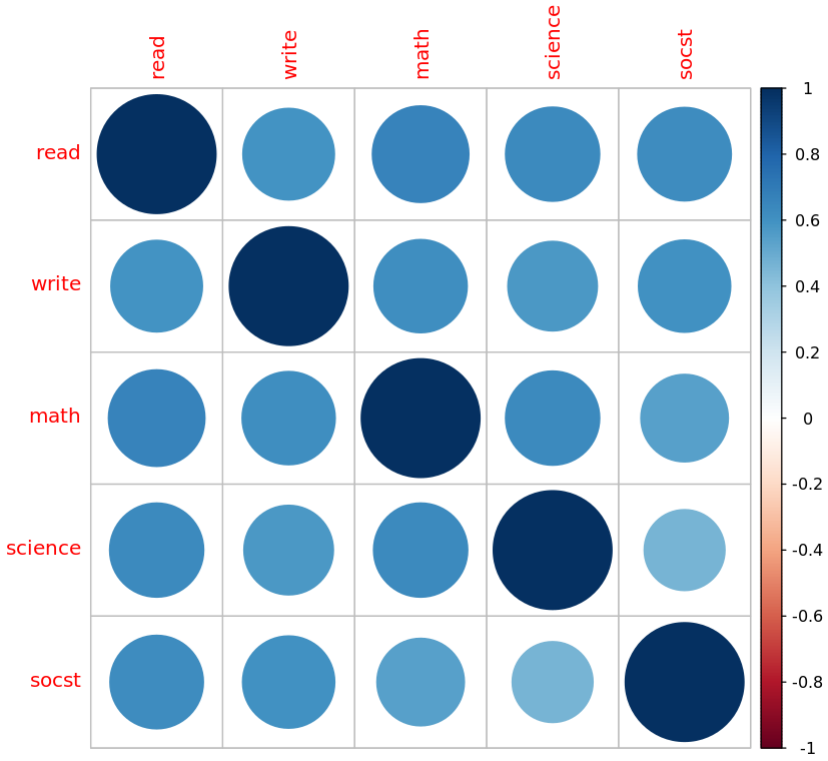
(c) Compute the correlation matrix for the five subject scores.

In [7]:
```
CM <- cor(data01[ , 6:10])
CM
```

A matrix: 5 × 5 of type dbl

	read	write	math	science	socst
read	1.0000000	0.5967765	0.6622801	0.6301579	0.6214843
write	0.5967765	1.0000000	0.6174493	0.5704416	0.6047932
math	0.6622801	0.6174493	1.0000000	0.6307332	0.5444803
science	0.6301579	0.5704416	0.6307332	1.0000000	0.4651060
socst	0.6214843	0.6047932	0.5444803	0.4651060	1.0000000

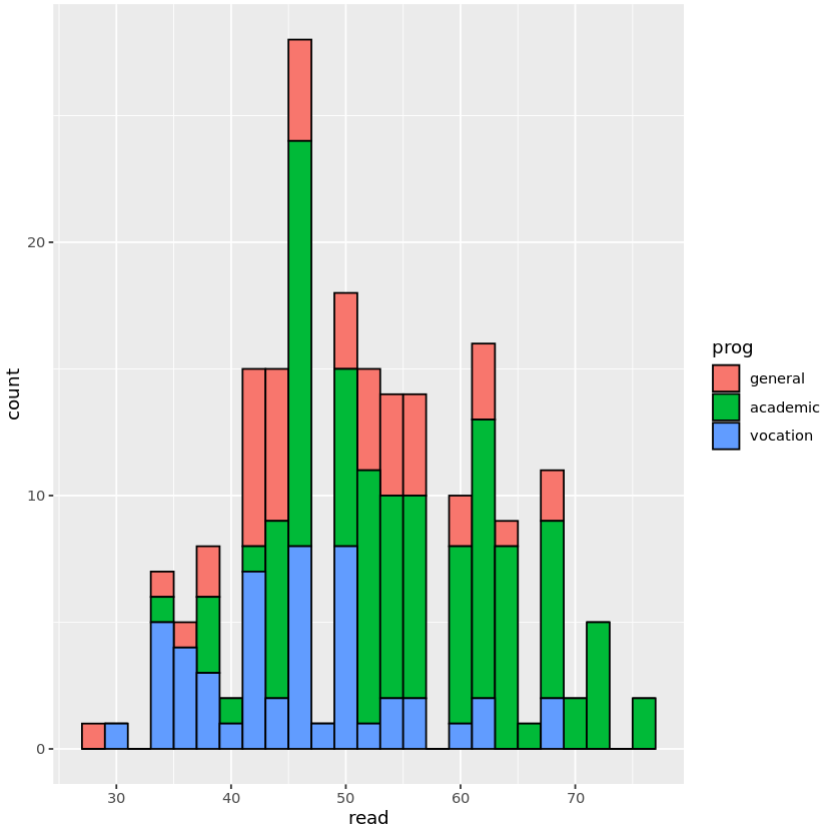
In [50]:
```
corrplot(CM)
corrplot(CM, 'number')
```



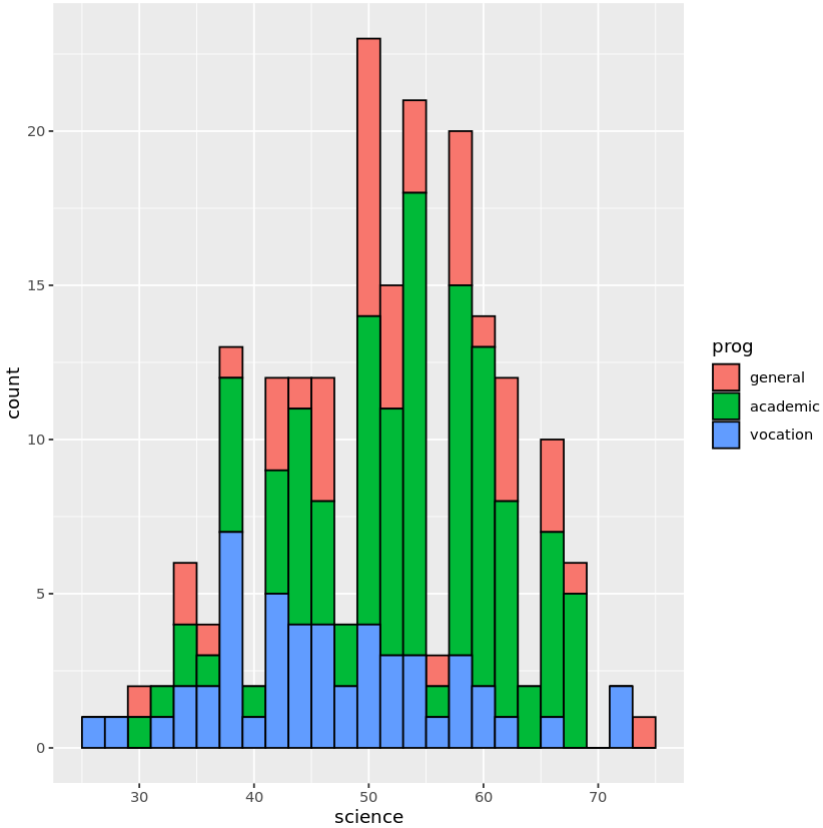
	read	write	math	science	socst
read	1.00	0.60	0.66	0.63	0.62
write	0.60	1.00	0.62	0.57	0.60
math	0.66	0.62	1.00	0.63	0.54
science	0.63	0.57	0.63	1.00	0.47
socst	0.62	0.60	0.54	0.47	1.00

In [9]:

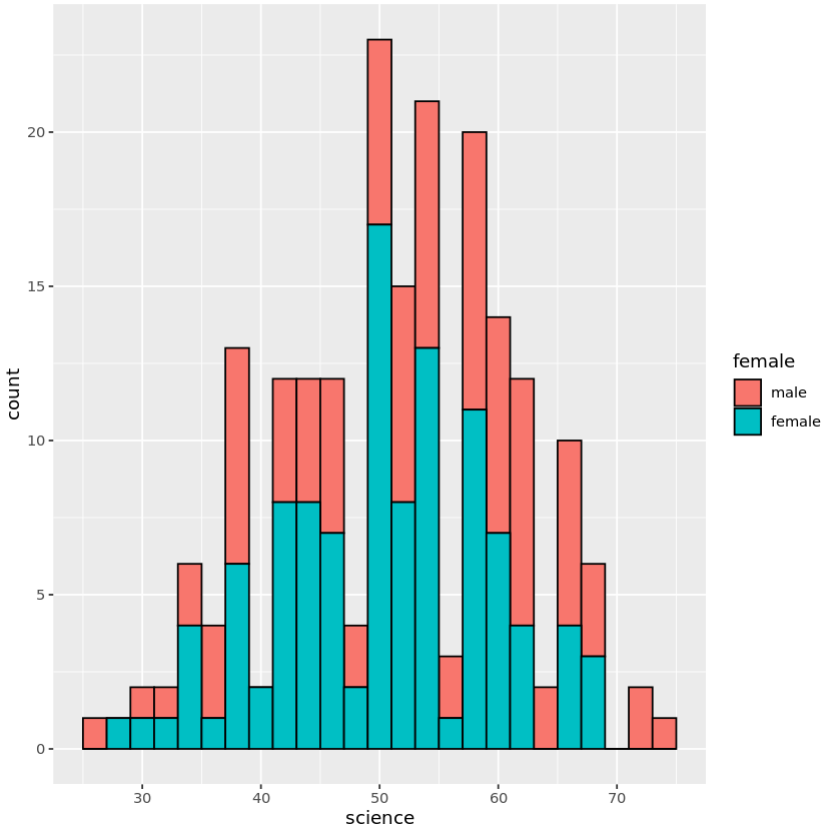
```
ggplot(data01, aes(read)) +
  geom_histogram(aes(fill = prog), color = "black", binwidth = 2)
```



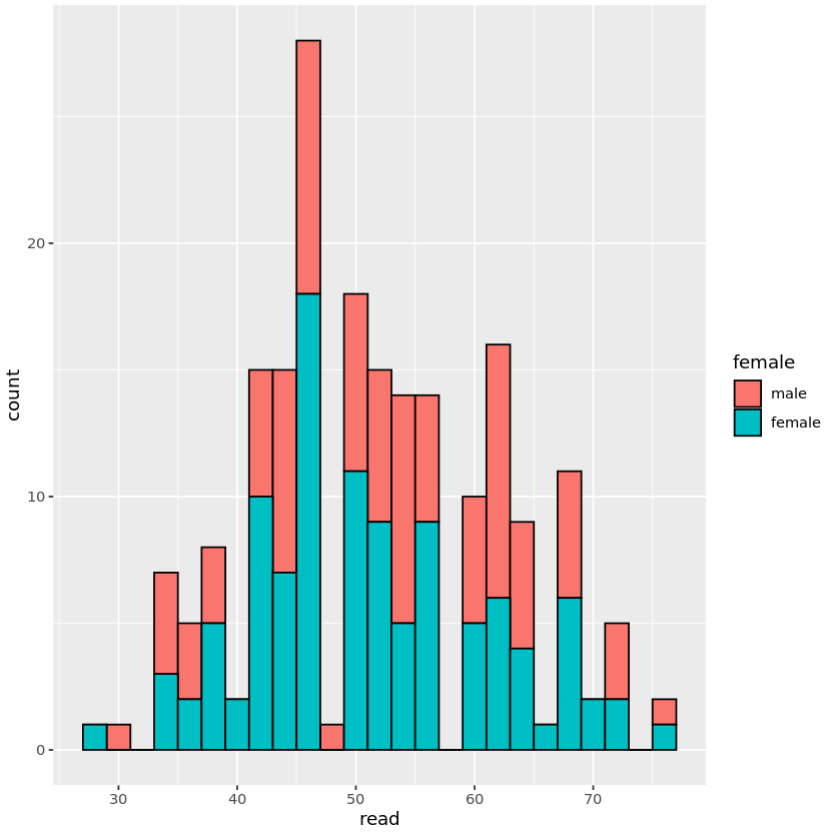
In [10]:
```
ggplot(data01, aes(science)) +
  geom_histogram(aes(fill = prog), color = "black", binwidth = 2)
```



In [11]:
```
ggplot(data01, aes(science)) +
  geom_histogram(aes(fill = female), color = "black", binwidth = 2)
```

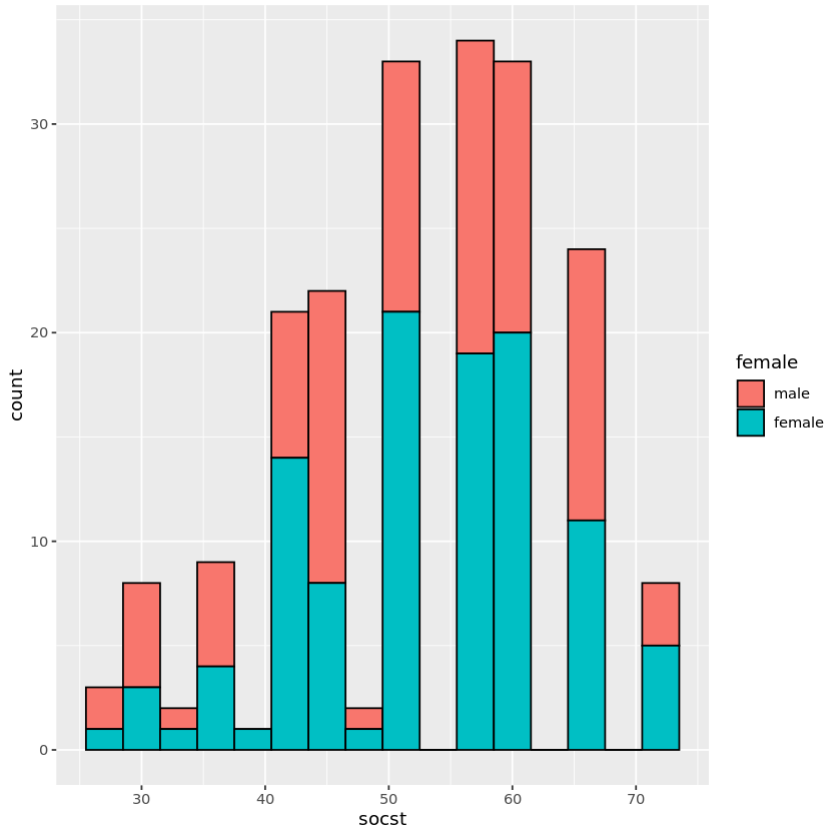


In [12]:
```
ggplot(data01, aes(read)) +
  geom_histogram(aes(fill = female), color = "black", binwidth = 2)
```



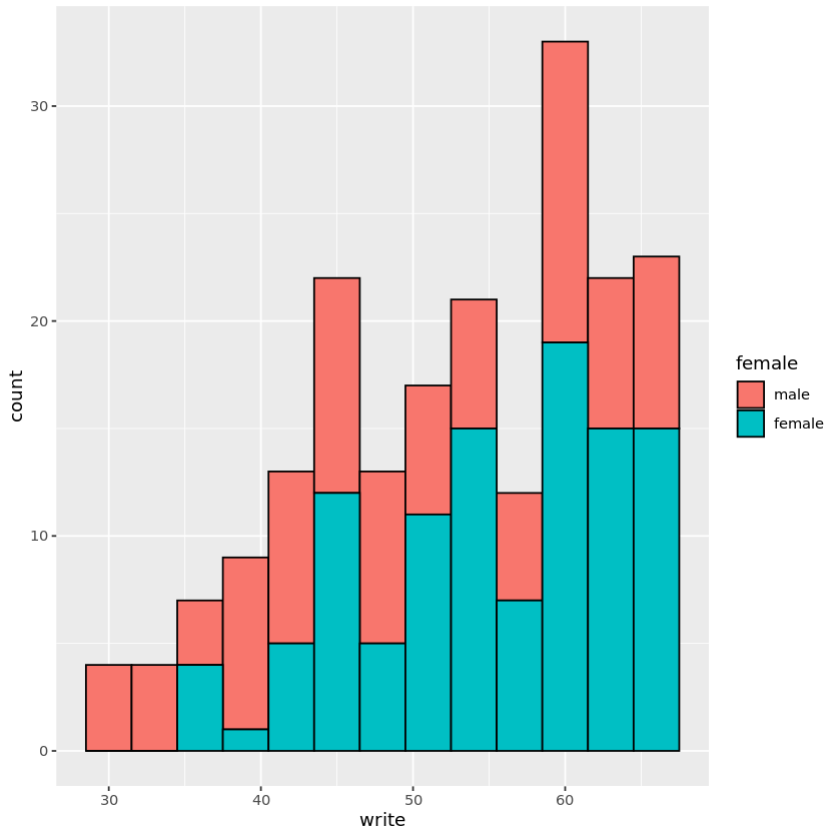
In [13]:

```
ggplot(data01, aes(socst)) +
  geom_histogram(aes(fill = female), color = "black", binwidth = 3)
```

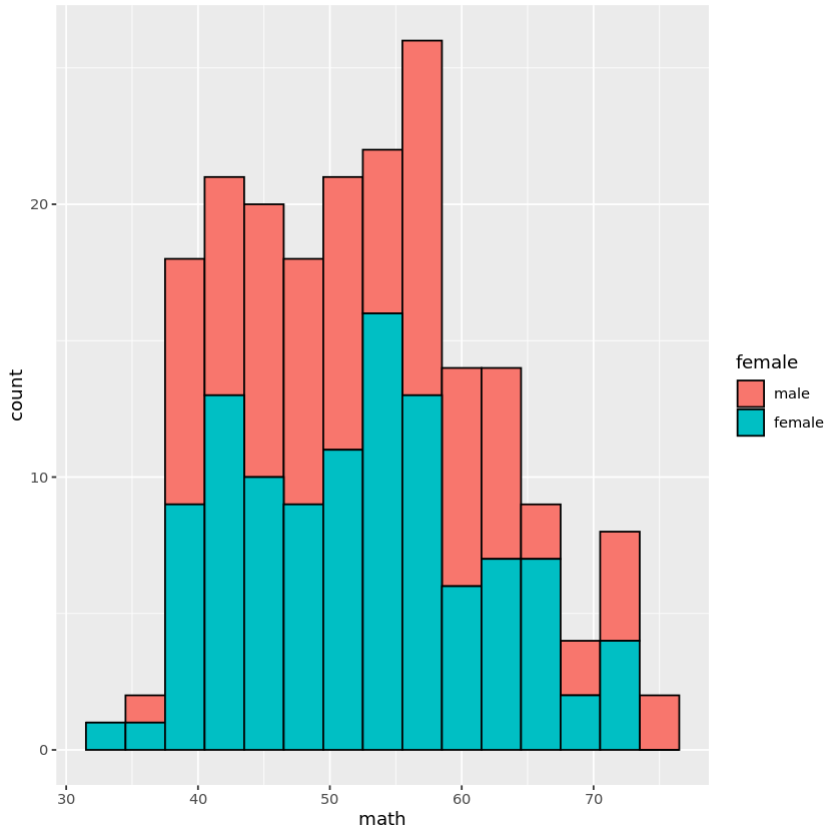


In [14]:

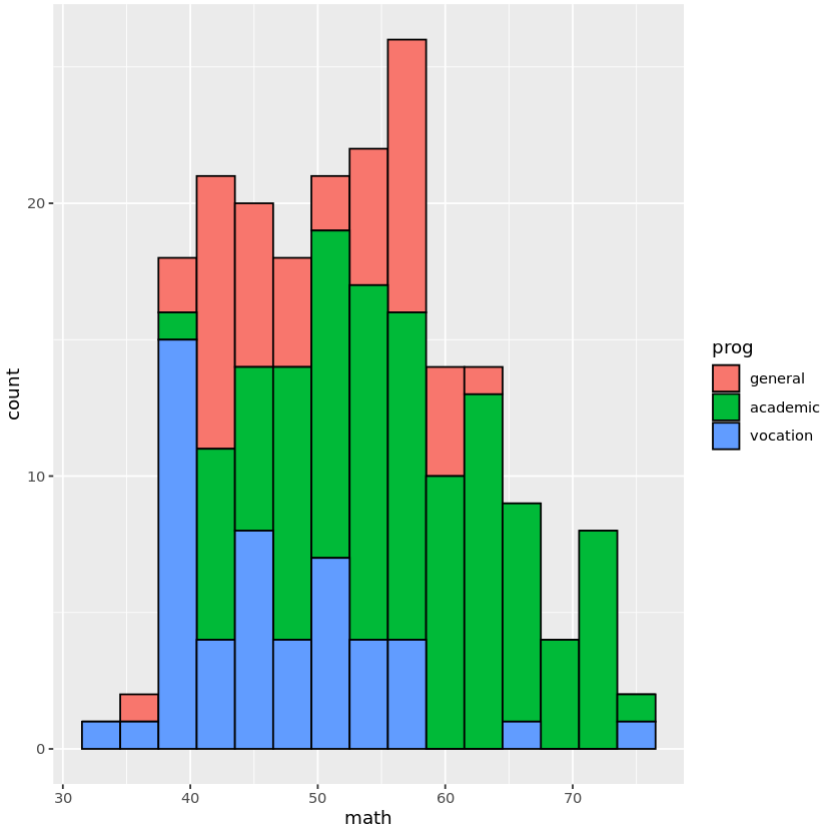
```
ggplot(data01, aes(write)) +
  geom_histogram(aes(fill = female), color = "black", binwidth = 3)
```



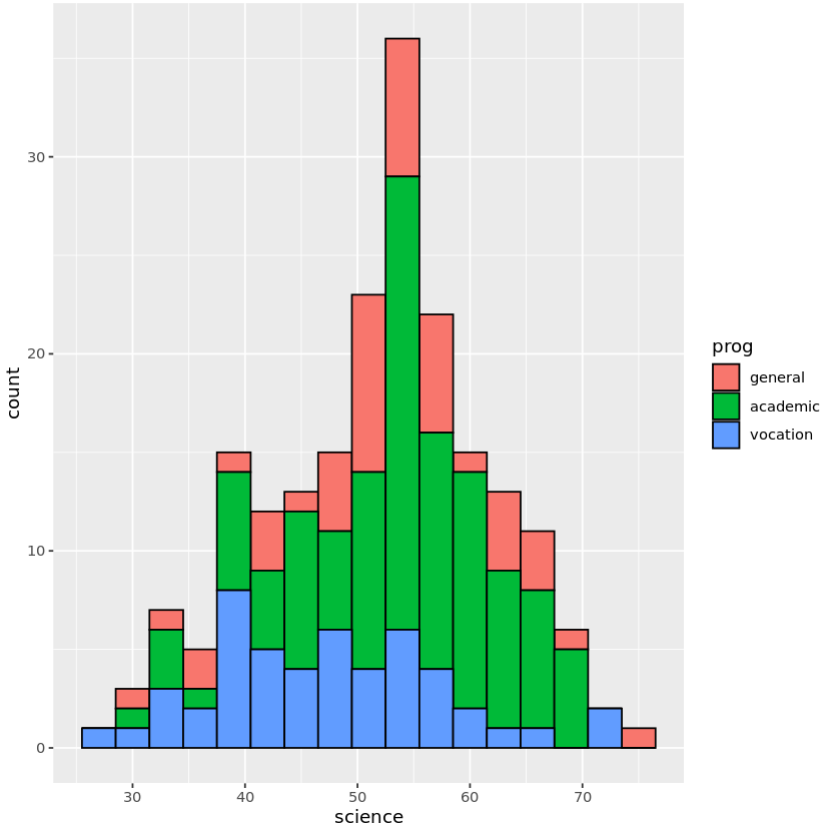
In [15]:
```
ggplot(data01, aes(math)) +
  geom_histogram(aes(fill = female), color = "black", binwidth = 3)
```



In [16]:
```
ggplot(data01, aes(math)) +
  geom_histogram(aes(fill = prog), color = "black", binwidth = 3)
```



In [17]:
```
ggplot(data01, aes(science)) +
  geom_histogram(aes(fill = prog), color = "black", binwidth = 3)
```



In [18]:
```
names(data01)
```

'id' · 'female' · 'ses' · 'schtyp' · 'prog' · 'read' · 'write' · 'math' · 'science' · 'socst' · 'honors' · 'awards' · 'cid'

In [19]:
```
levels(data01$ses)
```

'low' · 'middle' · 'high'

In [20]:
```
summary(data01)
```

```
      id            female        ses         schtyp          prog
Min.   :  1.00   male  : 91   low   :47   public :168   general : 45
1st Qu.: 50.75   female:109   middle:95   private: 32   academic:105
Median :100.50                high  :58                 vocation: 50
Mean   :100.50
3rd Qu.:150.25
Max.   :200.00
     read           write            math          science
Min.   :28.00   Min.   :31.00   Min.   :33.00   Min.   :26.00
1st Qu.:44.00   1st Qu.:45.75   1st Qu.:45.00   1st Qu.:44.00
Median :50.00   Median :54.00   Median :52.00   Median :53.00
Mean   :52.23   Mean   :52.77   Mean   :52.65   Mean   :51.85
3rd Qu.:60.00   3rd Qu.:60.00   3rd Qu.:59.00   3rd Qu.:58.00
Max.   :76.00   Max.   :67.00   Max.   :75.00   Max.   :74.00
    socst                honors        awards          cid
Min.   :26.00   not enrolled:147   Min.   :0.00   Min.   : 1.00
1st Qu.:46.00   enrolled    : 53   1st Qu.:0.00   1st Qu.: 5.00
Median :52.00                      Median :1.00   Median :10.50
Mean   :52.41                      Mean   :1.67   Mean   :10.43
3rd Qu.:61.00                      3rd Qu.:2.00   3rd Qu.:15.00
Max.   :71.00                      Max.   :7.00   Max.   :20.00
```

In [21]:
```
# Proportion of males and females choosing the three different programs:
require(formattable)
mf <- group_by(data01, female, prog) %>% summarise(count=n()) %>%
  group_by(prog) %>% mutate(etotal=sum(count), proportion=round(100*count/etotal,2))
```

```
Loading required package: formattable


Attaching package: ‘formattable’


The following object is masked from ‘package:MASS’:

    area


The following object is masked from ‘package:igraph’:

    normalize


`summarise()` has grouped output by 'female'. You can override using the `.groups` argume
nt.
```

In [22]:
```
mf
```

A grouped_df: 6 × 5

female	prog	count	etotal	proportion
<fct>	<fct>	<int>	<int>	<dbl>
male	general	21	45	46.67
male	academic	47	105	44.76
male	vocation	23	50	46.00
female	general	24	45	53.33
female	academic	58	105	55.24
female	vocation	27	50	54.00

In [23]:
```
ss <- group_by(data01, ses, female, prog) %>% summarise(count=n()) %>%
  group_by(prog) %>% mutate(etotal=sum(count), proportion=round(100*count/etotal,2))
```

```
`summarise()` has grouped output by 'ses', 'female'. You can override using the `.groups`
argument.
```

In [24]:
```
ss
```

A grouped_df: 18 × 6

ses	female	prog	count	etotal	proportion
<fct>	<fct>	<fct>	<int>	<int>	<dbl>
low	male	general	7	45	15.56
low	male	academic	4	105	3.81
low	male	vocation	4	50	8.00
low	female	general	9	45	20.00
low	female	academic	15	105	14.29
low	female	vocation	8	50	16.00
middle	male	general	10	45	22.22
middle	male	academic	22	105	20.95
middle	male	vocation	15	50	30.00
middle	female	general	10	45	22.22
middle	female	academic	22	105	20.95
middle	female	vocation	16	50	32.00
high	male	general	4	45	8.89
high	male	academic	21	105	20.00
high	male	vocation	4	50	8.00
high	female	general	5	45	11.11
high	female	academic	21	105	20.00

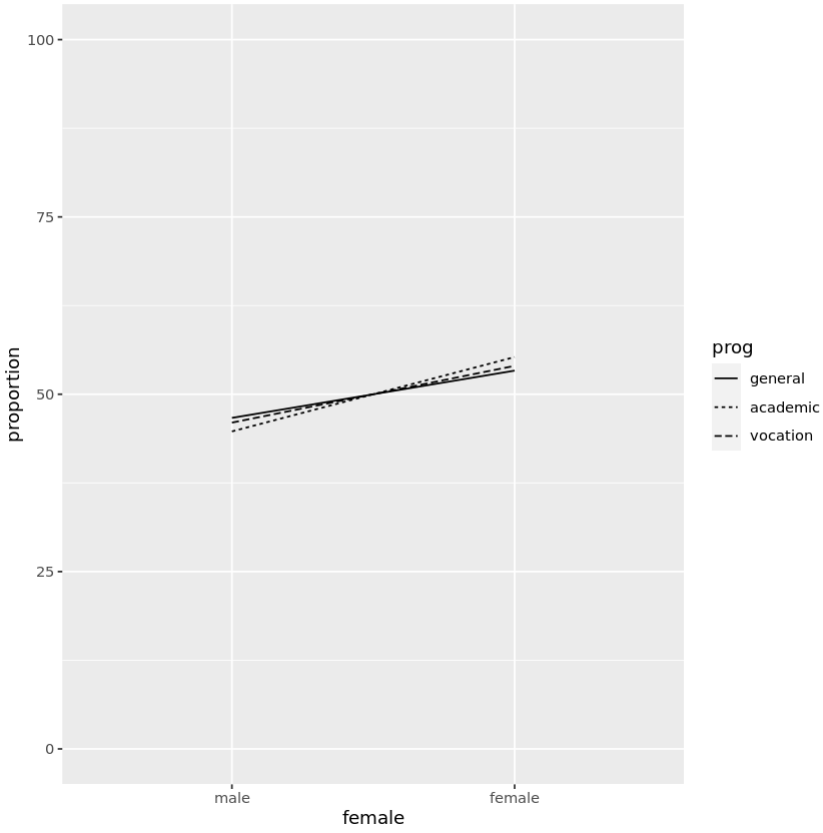
ses	female	prog	count	etotal	proportion
<fct>	<fct>	<fct>	<int>	<int>	<dbl>
high	female	vocation	3	50	6.00

In [25]:
```
#install.packages('formattable')
library(formattable)
formattable(mf)
```

A formattable: 6 × 5

female	prog	count	etotal	proportion
<fct>	<fct>	<int>	<int>	<dbl>
male	general	21	45	46.67
male	academic	47	105	44.76
male	vocation	23	50	46.00
female	general	24	45	53.33
female	academic	58	105	55.24
female	vocation	27	50	54.00

In [26]:
```
# Visualize the data using GGPLOT:
ggplot(mf, aes(x=female, y=proportion, group=prog, linetype=prog)) +ylim(0,100) + geom_li
```



In [27]:
```
# Visualize the data using DPLYR:
# Proportion by Income Group
```

```
readstat <- mutate(data01, readgp=cut_number(read, 8)) %>% group_by(readgp, prog) %>%
  summarise(count=n()) %>% group_by(readgp) %>%
  mutate(etotal=sum(count), proportion=count/etotal)
```

```
`summarise()` has grouped output by 'readgp'. You can override using the `.groups` argume
nt.
```

In [28]:

```
readstat
```

A grouped_df: 24 × 5

readgp	prog	count	etotal	proportion
<fct>	<fct>	<int>	<int>	<dbl>
[28,42]	general	11	37	0.29729730
[28,42]	academic	6	37	0.16216216
[28,42]	vocation	20	37	0.54054054
(42,44]	general	7	15	0.46666667
(42,44]	academic	5	15	0.33333333
(42,44]	vocation	3	15	0.20000000
(44,47]	general	4	30	0.13333333
(44,47]	academic	18	30	0.60000000
(44,47]	vocation	8	30	0.26666667
(47,50]	general	3	19	0.15789474
(47,50]	academic	7	19	0.36842105
(47,50]	vocation	9	19	0.47368421
(50,55]	general	8	29	0.27586207
(50,55]	academic	18	29	0.62068966
(50,55]	vocation	3	29	0.10344828
(55,60]	general	6	23	0.26086957
(55,60]	academic	14	23	0.60869565
(55,60]	vocation	3	23	0.13043478
(60,65]	general	4	26	0.15384615
(60,65]	academic	20	26	0.76923077
(60,65]	vocation	2	26	0.07692308
(65,76]	general	2	21	0.09523810
(65,76]	academic	17	21	0.80952381
(65,76]	vocation	2	21	0.09523810

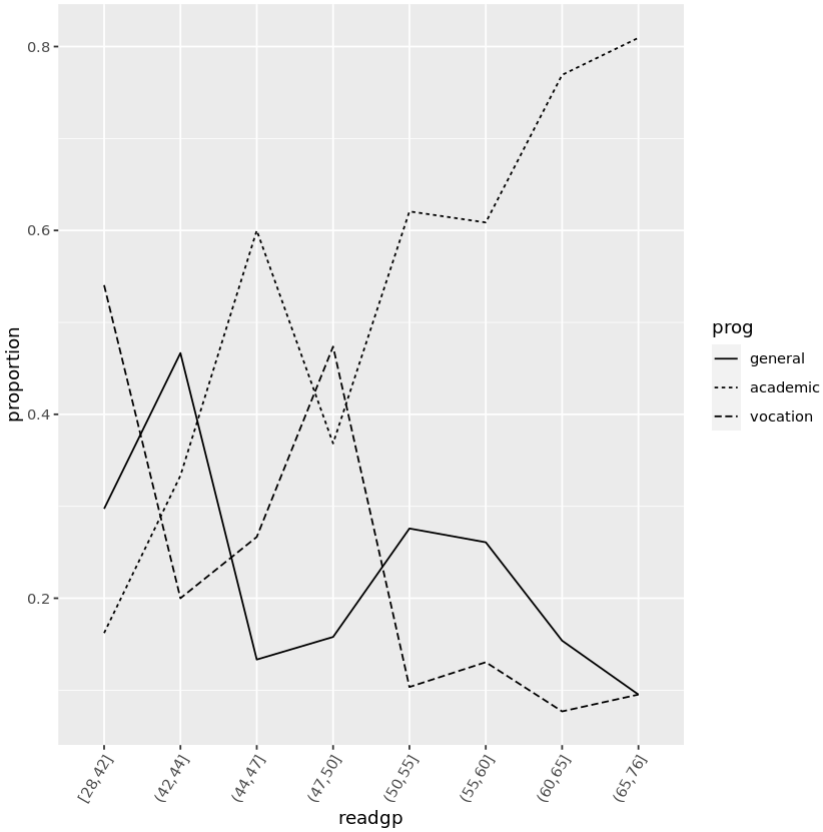
In [29]:

```
formattable(readstat)
```

A formattable: 24 × 5

readgp	prog	count	etotal	proportion
<fct>	<fct>	<int>	<int>	<dbl>
[28,42]	general	11	37	0.29729730
[28,42]	academic	6	37	0.16216216
[28,42]	vocation	20	37	0.54054054
(42,44]	general	7	15	0.46666667
(42,44]	academic	5	15	0.33333333
(42,44]	vocation	3	15	0.20000000
(44,47]	general	4	30	0.13333333
(44,47]	academic	18	30	0.60000000
(44,47]	vocation	8	30	0.26666667
(47,50]	general	3	19	0.15789474
(47,50]	academic	7	19	0.36842105
(47,50]	vocation	9	19	0.47368421
(50,55]	general	8	29	0.27586207
(50,55]	academic	18	29	0.62068966
(50,55]	vocation	3	29	0.10344828
(55,60]	general	6	23	0.26086957
(55,60]	academic	14	23	0.60869565
(55,60]	vocation	3	23	0.13043478
(60,65]	general	4	26	0.15384615
(60,65]	academic	20	26	0.76923077
(60,65]	vocation	2	26	0.07692308
(65,76]	general	2	21	0.09523810
(65,76]	academic	17	21	0.80952381
(65,76]	vocation	2	21	0.09523810

In [30]:
```
ggplot(readstat, aes(x=readgp, y=proportion, group=prog, linetype=prog)) + geom_line()+ t
```



In [31]:
```
# Visualize the data using DPLYR:
# Proportion by Income Group
with(data01, table(ses, prog))
```

```
        prog
ses      general academic vocation
  low         16       19       12
  middle      20       44       31
  high         9       42        7
```

In [32]:
```
with(data01, do.call(rbind, tapply(write, prog, function(x) c(M = mean(x), SD = sd(x)))))
```

A matrix: 3 × 2 of type dbl

	M	SD
general	51.33333	9.397775
academic	56.25714	7.943343
vocation	46.76000	9.318754

In the help file the ddply function call should say "summarise" instead of "summarize". Otherwise get the error message: "Error: argument "by" is missing, with no default"

In [33]:
```
formattable(data01 %>% group_by(prog) %>% summarise(n=n(), M = mean(write), SD = sd(write
```

A formattable: 3 × 4

prog	n	M	SD
<fct>	**<int>**	**<dbl>**	**<dbl>**

prog	n	M	SD
<fct>	<int>	<dbl>	<dbl>
general	45	51.33333	9.397775
academic	105	56.25714	7.943343
vocation	50	46.76000	9.318754

In [34]:
```
# Proportion of students by socio economic status
socstat <- group_by(data01, female, ses, prog) %>% summarise(count=n()) %>%
  group_by(prog) %>% mutate(etotal=sum(count), proportion=round(100*count/etotal,2))
```

```
`summarise()` has grouped output by 'female', 'ses'. You can override using the `.groups`
argument.
```

In [35]:
```
socstat
```

A grouped_df: 18 × 6

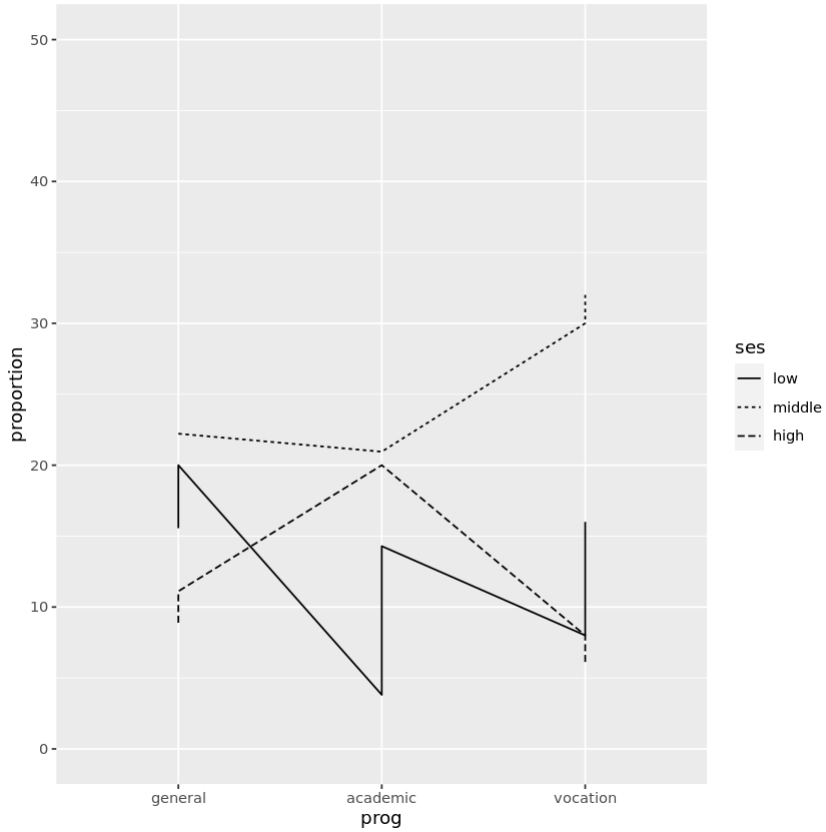
female	ses	prog	count	etotal	proportion
<fct>	<fct>	<fct>	<int>	<int>	<dbl>
male	low	general	7	45	15.56
male	low	academic	4	105	3.81
male	low	vocation	4	50	8.00
male	middle	general	10	45	22.22
male	middle	academic	22	105	20.95
male	middle	vocation	15	50	30.00
male	high	general	4	45	8.89
male	high	academic	21	105	20.00
male	high	vocation	4	50	8.00
female	low	general	9	45	20.00
female	low	academic	15	105	14.29
female	low	vocation	8	50	16.00
female	middle	general	10	45	22.22
female	middle	academic	22	105	20.95
female	middle	vocation	16	50	32.00
female	high	general	5	45	11.11
female	high	academic	21	105	20.00
female	high	vocation	3	50	6.00

In [36]:
```
formattable(socstat)
```

A formattable: 18 × 6

female	ses	prog	count	etotal	proportion
<fct>	<fct>	<fct>	<int>	<int>	<dbl>
male	low	general	7	45	15.56
male	low	academic	4	105	3.81
male	low	vocation	4	50	8.00
male	middle	general	10	45	22.22
male	middle	academic	22	105	20.95
male	middle	vocation	15	50	30.00
male	high	general	4	45	8.89
male	high	academic	21	105	20.00
male	high	vocation	4	50	8.00
female	low	general	9	45	20.00
female	low	academic	15	105	14.29
female	low	vocation	8	50	16.00
female	middle	general	10	45	22.22
female	middle	academic	22	105	20.95
female	middle	vocation	16	50	32.00
female	high	general	5	45	11.11
female	high	academic	21	105	20.00
female	high	vocation	3	50	6.00

In [37]:
```
ggplot(socstat, aes(x=prog, y=proportion, group=ses, linetype=ses)) +ylim(0,50) + geom_li
```



(d) Fit a multinomial response model for the program choice and examine the fitted coefficients. Of the five subjects, one gives unexpected coefficients. Identify this subject and suggest an explanation for this behavior.

In [38]:
```
# fit the Multinomial Model:
# Reference point
data01$prog2 <- relevel(data01$prog, ref = "academic")
data01$prog2
```

vocation · general · vocation · vocation · vocation · general · vocation · vocation · vocation · vocation · vocation · academic · vocation · vocation · vocation · general · general · vocation · academic · vocation · general · vocation · vocation · vocation · academic · academic · general · general · academic · academic · general · vocation · academic · academic · vocation · vocation · vocation · academic · general · academic · general · academic · academic · vocation · academic · vocation · vocation · general · vocation · academic · academic · vocation · general · academic · academic · general · academic · general · vocation · general · vocation · academic · academic · vocation · vocation · vocation · general · academic · academic · general · academic · academic · academic · general · vocation · general · vocation · general · general · academic · vocation · academic · academic · general · vocation · academic · general · general · general · vocation · vocation · general · vocation · academic · vocation · general · academic · vocation · vocation · general · academic · vocation · vocation · vocation · general · academic · general · general · academic · academic · academic · academic · academic · academic · academic · vocation · academic · academic · academic · academic · vocation · academic · academic · academic · academic · academic · academic · academic · academic · academic · academic · general · academic · academic · general · academic · academic · academic · academic · academic · academic · general · general · academic · general · general · general · academic · academic · academic · academic · academic · vocation · general · academic · academic · academic · academic · academic · general · general · academic · academic · academic · vocation · general · academic ·

academic · general · general · academic · academic · vocation · academic · academic · academic · academic · academic · academic · academic · academic · academic · general · academic · academic · academic · academic · academic · academic · academic · academic · academic · academic · academic · vocation · academic · academic · academic

▶ **Levels**:

In [39]:
```
# Fit Multinomial Regression
test <- multinom(prog2 ~ ses + write, data = data01)
summary(test)
```

```
# weights:  15 (8 variable)
initial  value 219.722458
iter  10 value 179.982880
final  value 179.981726
converged
Call:
multinom(formula = prog2 ~ ses + write, data = data01)

Coefficients:
         (Intercept)  sesmiddle    seshigh      write
general     2.852198 -0.5332810 -1.1628226 -0.0579287
vocation    5.218260  0.2913859 -0.9826649 -0.1136037

Std. Errors:
         (Intercept) sesmiddle   seshigh      write
general     1.166441 0.4437323 0.5142196 0.02141097
vocation    1.163552 0.4763739 0.5955665 0.02221996

Residual Deviance: 359.9635
AIC: 375.9635
```

In [51]:
```
## Fit predictions based on model
data01_probs <- predict (test, data01, "probs") # predict on new data
data01_probs[1:200]
data01_pred <- predict (test, data01)
data01_pred[1:5]
table(data01_pred,data01$prog)
```

0.148276434691488 · 0.120201734814452 · 0.418674745525916 · 0.172688536345951 · 0.100123137365158 · 0.353356571988496 · 0.156256213894361 · 0.100123137365158 · 0.233129192891448 · 0.169940240230941 · 0.277772709260425 · 0.291750166218769 · 0.107168677245435 · 0.288877944632198 · 0.148276434691488 · 0.277772709260425 · 0.312625087608512 · 0.329389753055937 · 0.329389753055937 · 0.632459800219732 · 0.199858253564114 · 0.288877944632198 · 0.330660885514585 · 0.277772709260425 · 0.172688536345951 · 0.396633316585577 · 0.367693485438978 · 0.288877944632198 · 0.229200688917725 · 0.386578675076207 · 0.288877944632198 · 0.19969105900606 · 0.214141008540118 · 0.312625087608512 · 0.233129192891448 · 0.199858253564114 · 0.120201734814452 · 0.792140722362904 · 0.229200688917725 · 0.350453175182767 · 0.100123137365158 · 0.485293573939272 · 0.482354103967011 · 0.440861245139527 · 0.277772709260425 · 0.463214074466187 · 0.329389753055937 · 0.463214074466187 · 0.250970557331844 · 0.393791009628624 · 0.233129192891448 · 0.233129192891448 ·

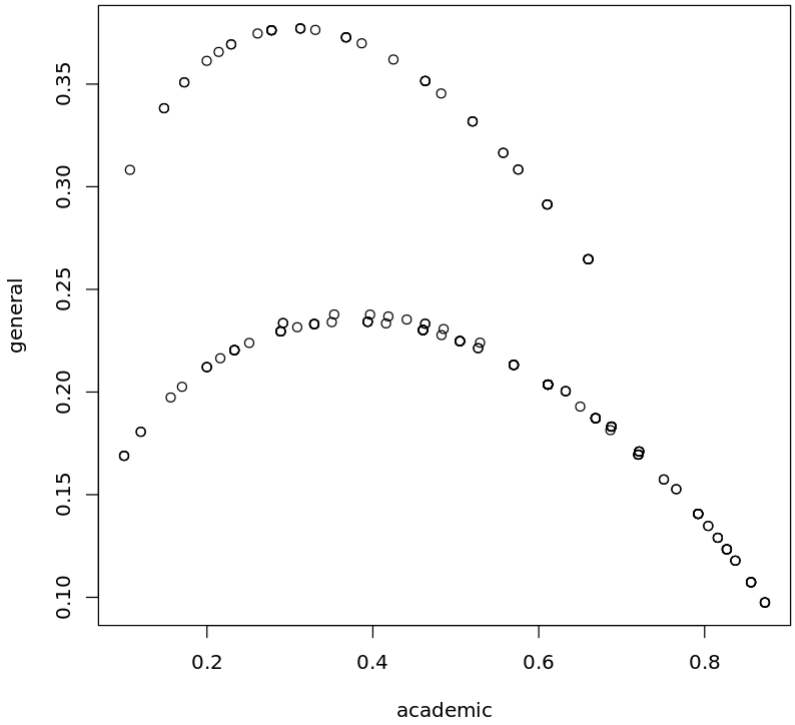
0.277772709260425 · 0.520221527748053 · 0.460465180315754 · 0.367693485438978 · 0.463098641274867 · 0.765782185639655 · 0.610277399381601 · 0.291750166218769 · 0.504926954356892 · 0.216082540346683 · 0.687632169297203 · 0.569842516641632 · 0.393791009628624 · 0.329389753055937 · 0.520221527748053 · 0.393791009628624 · 0.460465180315754 · 0.482757109944354 · 0.721014022087381 · 0.569842516641632 · 0.463098641274867 · 0.424795523966963 · 0.569842516641632 · 0.463214074466187 · 0.460465180315754 · 0.460465180315754 · 0.288877944632198 · 0.329389753055937 · 0.367693485438978 · 0.504926954356892 · 0.687632169297203 · 0.277772709260425 · 0.632459800219732 · 0.312625087608512 · 0.504926954356892 · 0.199858253564114 · 0.611036666154868 · 0.308840034781615 · 0.687632169297203 · 0.721014022087381 · 0.529196046009234 · 0.415887182873568 · 0.66853751214223 · 0.611036666154868 · 0.460465180315754 · 0.460465180315754 · 0.329389753055937 · 0.233129192891448 · 0.460465180315754 · 0.460465180315754 · 0.526885928045748 · 0.233129192891448 · 0.393791009628624 · 0.55720577449979 · 0.463214074466187 · 0.504926954356892 · 0.611036666154868 · 0.526885928045748 · 0.610277399381601 · 0.504926954356892 · 0.504926954356892 · 0.504926954356892 · 0.611036666154868 · 0.569842516641632 · 0.81583762466641 · 0.687632169297203 · 0.611036666154868 · 0.66853751214223 · 0.575254838964255 · 0.611036666154868 · 0.720034873890803 · 0.81583762466641 · 0.611036666154868 · 0.792140722362904 · 0.611036666154868 · 0.611036666154868 · 0.720034873890803 · 0.765782185639655 · 0.611036666154868 · 0.632459800219732 · 0.367693485438978 · 0.792140722362904 · 0.66853751214223 · 0.792140722362904 · 0.504926954356892 · 0.837033162668371 · 0.261048421918382 · 0.721014022087381 · 0.687632169297203 · 0.520221527748053 · 0.569842516641632 · 0.569842516641632 · 0.65000105498486 · 0.66853751214223 · 0.66853751214223 · 0.659679152104757 · 0.569842516641632 · 0.792140722362904 · 0.611036666154868 · 0.826736856046208 · 0.720034873890803 · 0.66853751214223 · 0.610277399381601 · 0.66853751214223 · 0.721014022087381 · 0.826736856046208 · 0.659679152104757 · 0.826736856046208 · 0.872677228682114 · 0.855913552529764 · 0.575254838964255 · 0.792140722362904 · 0.611036666154868 · 0.611036666154868 · 0.855913552529764 · 0.826736856046208 · 0.855913552529764 · 0.792140722362904 · 0.55720577449979 · 0.81583762466641 · 0.659679152104757 · 0.872677228682114 · 0.855913552529764 · 0.659679152104757 · 0.750814034588517 · 0.659679152104757 · 0.610277399381601 · 0.687632169297203 · 0.837033162668371 · 0.611036666154868 · 0.855913552529764 · 0.611036666154868 · 0.872677228682114 · 0.872677228682114 · 0.804312653556822 · 0.872677228682114 · 0.826736856046208 · 0.504926954356892 · 0.855913552529764 · 0.826736856046208 · 0.611036666154868 · 0.804312653556822 · 0.837033162668371 · 0.855913552529764 · 0.686401689882404 · 0.750814034588517 · 0.720034873890803 · 0.66853751214223

vocation · vocation · academic · vocation · vocation

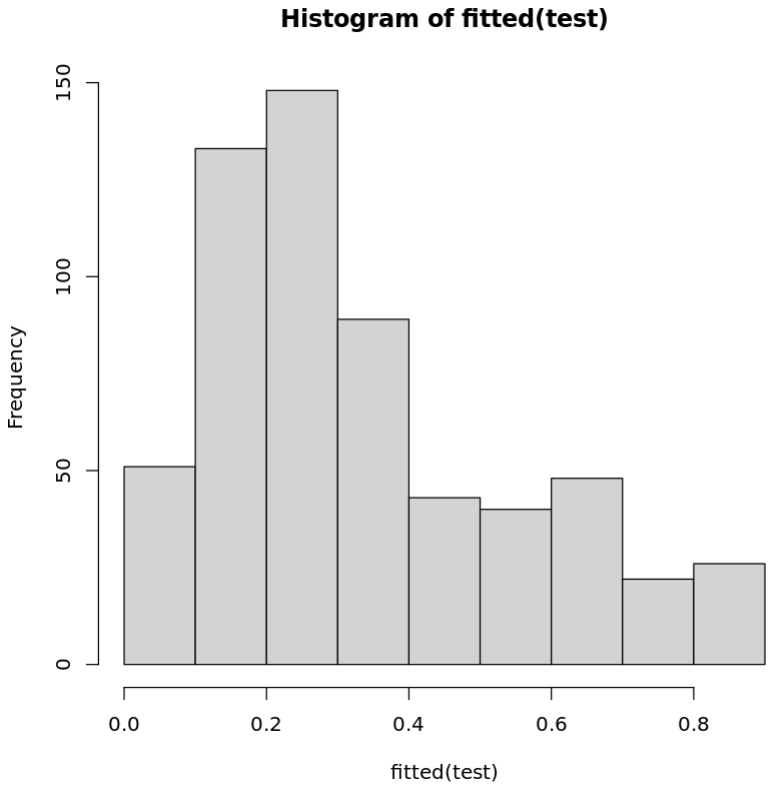
▶ **Levels**:

```
data01_pred general academic vocation
   academic      27       92       23
   general        7        4        4
   vocation      11        9       23
```

In [57]:
```
## Visuals for Predictions
plot(data01_probs)
```



In [53]:
```
hist(fitted(test))
```



In [40]:
```
# Fit Multinomial Regression
test_all <- multinom(prog2 ~ ses + write + read + math + science + socst, data = data01)
summary(test_all)
```

```
# weights:  27 (16 variable)
initial  value 219.722458
iter  10 value 181.523639
iter  20 value 159.901319
final  value 159.901300
converged
Call:
multinom(formula = prog2 ~ ses + write + read + math + science +
    socst, data = data01)

Coefficients:
         (Intercept)  sesmiddle    seshigh       write        read        math
general     4.441175 -0.4075732 -1.0154317 -0.02638071 -0.04451857 -0.09692786
vocation    9.239398  0.8510902 -0.3106137 -0.03414681 -0.03452593 -0.11706045
            science       socst
general  0.09983693 -0.02169398
vocation 0.05430606 -0.07079844

Std. Errors:
         (Intercept) sesmiddle   seshigh      write       read       math
general     1.520336  0.492940 0.5762707 0.03009256 0.03011511 0.03359206
vocation    1.678261  0.546589 0.6639833 0.03019865 0.03345737 0.03683209
            science      socst
general  0.03020831 0.02622514
vocation 0.03105560 0.02737621

Residual Deviance: 319.8026
AIC: 351.8026
```

Of the five subjects, one gives unexpected coefficients: science - they are positive. Identify this subject and suggest an explanation for this behavior.....(e) Construct a derived variable that is the sum of the five subject scores. Fit a multinomial model as before except with this one sum variable in place of the five subjects separately. Compare the two models to decide which should be preferred.

In [54]:
```
SUMscore <- data01$read + data01$write + data01$math + data01$science + data01$socst
SUMscore
test_sc <- multinom(prog2 ~ SUMscore, data = data01)
summary(test_sc)
```

```
165 · 180 · 190 · 181 · 200 · 190 · 198 · 186 · 196 · 187 · 198 · 198 · 194 · 200 · 194 · 207 · 212 · 208 ·
200 · 220 · 206 · 211 · 218 · 199 · 215 · 216 · 223 · 224 · 215 · 225 · 211 · 201 · 206 · 216 · 201 · 222 ·
212 · 227 · 217 · 228 · 228 · 239 · 219 · 225 · 227 · 220 · 240 · 215 · 231 · 231 · 236 · 221 · 232 · 239 ·
234 · 240 · 240 · 226 · 234 · 220 · 236 · 246 · 251 · 241 · 244 · 239 · 237 · 255 · 250 · 237 · 248 · 237 ·
253 · 254 · 258 · 253 · 253 · 254 · 245 · 260 · 236 · 241 · 256 · 241 · 248 · 268 · 264 · 250 · 265 · 255 ·
241 · 253 · 263 · 266 · 239 · 264 · 270 · 266 · 271 · 262 · 271 · 266 · 272 · 243 · 254 · 279 · 265 · 265 ·
275 · 265 · 276 · 268 · 273 · 269 · 264 · 265 · 280 · 270 · 276 · 272 · 258 · 288 · 278 · 253 · 268 · 283 ·
284 · 276 · 286 · 287 · 297 · 282 · 273 · 293 · 273 · 289 · 295 · 275 · 274 · 291 · 291 · 287 · 288 · 283 ·
288 · 294 · 299 · 284 · 274 · 290 · 285 · 297 · 297 · 287 · 287 · 278 · 294 · 295 · 295 · 305 · 300 · 307 ·
308 · 293 · 314 · 304 · 310 · 310 · 307 · 312 · 292 · 312 · 304 · 305 · 311 · 306 · 321 · 323 · 313 · 317 ·
317 · 308 · 323 · 325 · 326 · 327 · 327 · 332 · 327 · 329 · 334 · 325 · 320 · 336 · 332 · 339 · 339 · 340 ·
330 · 343
```

```
# weights:  9 (4 variable)
initial  value 219.722458
iter  10 value 176.619114
iter  10 value 176.619114
final  value 176.619114
converged
Call:
multinom(formula = prog2 ~ SUMscore, data = data01)

Coefficients:
         (Intercept)    SUMscore
general     4.191407 -0.01886820
vocation    8.645141 -0.03679921

Std. Errors:
         (Intercept)    SUMscore
general     1.367874 0.005157952
vocation    1.505506 0.006007857

Residual Deviance: 353.2382
AIC: 361.2382
```

(f) Use a stepwise method to reduce the model. Which variables are in your selected model?

In [55]:
```
step(test)
```

```
Start:  AIC=375.96
prog2 ~ ses + write

trying - ses
# weights:  9 (4 variable)
initial  value 219.722458
final  value 185.510837
converged
trying - write
# weights:  12 (6 variable)
initial  value 219.722458
iter  10 value 195.705189
iter  10 value 195.705188
iter  10 value 195.705188
final  value 195.705188
converged
        Df      AIC
<none>   8 375.9635
- ses    4 379.0217
- write  6 403.4104
Call:
multinom(formula = prog2 ~ ses + write, data = data01)

Coefficients:
         (Intercept)  sesmiddle    seshigh      write
general     2.852198 -0.5332810 -1.1628226 -0.0579287
vocation    5.218260  0.2913859 -0.9826649 -0.1136037

Residual Deviance: 359.9635
AIC: 375.9635
```

In [41]:
```
stepwise_model <- test_all %>% stepAIC(trace = FALSE)
coef(stepwise_model)
```

```
# weights:  21 (12 variable)
initial  value 219.722458
iter  10 value 173.294170
final  value 164.975567
converged
# weights:  24 (14 variable)
initial  value 219.722458
iter  10 value 184.270328
iter  20 value 160.624525
final  value 160.624514
converged
# weights:  24 (14 variable)
initial  value 219.722458
iter  10 value 189.441467
iter  20 value 161.107834
final  value 161.107830
converged
# weights:  24 (14 variable)
initial  value 219.722458
iter  10 value 192.869507
iter  20 value 167.071479
iter  20 value 167.071479
iter  20 value 167.071479
final  value 167.071479
converged
# weights:  24 (14 variable)
initial  value 219.722458
iter  10 value 193.222859
iter  20 value 165.985643
final  value 165.985637
converged
# weights:  24 (14 variable)
initial  value 219.722458
iter  10 value 187.735608
iter  20 value 163.564115
final  value 163.564109
converged
# weights:  24 (14 variable)
initial  value 219.722458
iter  10 value 184.270328
iter  20 value 160.624525
final  value 160.624514
converged
# weights:  18 (10 variable)
initial  value 219.722458
iter  10 value 173.157560
final  value 165.801985
converged
# weights:  21 (12 variable)
initial  value 219.722458
iter  10 value 171.569774
final  value 162.019070
converged
# weights:  21 (12 variable)
initial  value 219.722458
iter  10 value 176.347675
final  value 169.541007
converged
# weights:  21 (12 variable)
initial  value 219.722458
iter  10 value 176.109306
```

```
final  value 166.309153
converged
# weights:  21 (12 variable)
initial  value 219.722458
iter  10 value 181.721058
iter  20 value 165.864397
iter  20 value 165.864396
iter  20 value 165.864396
final  value 165.864396
converged
# weights:  21 (12 variable)
initial  value 219.722458
iter  10 value 171.569774
final  value 162.019070
converged
# weights:  15 (8 variable)
initial  value 219.722458
iter  10 value 167.561494
final  value 167.337045
converged
# weights:  18 (10 variable)
initial  value 219.722458
iter  10 value 175.180296
final  value 174.575922
converged
# weights:  18 (10 variable)
initial  value 219.722458
iter  10 value 167.437335
final  value 166.627692
converged
# weights:  18 (10 variable)
initial  value 219.722458
iter  10 value 170.088834
final  value 169.576379
converged
```

A matrix: 2 × 6 of type dbl

	(Intercept)	sesmiddle	seshigh	math	science	socst
general	3.925130	-0.3251216	-0.9441242	-0.1184824	0.08012764	-0.04237059
vocation	8.745662	0.9667470	-0.2018126	-0.1386983	0.03516103	-0.09091674

The function chose a final model in which two variables have been removed from the original full model. Dropped predictors are: read & write

In [42]:
```
summary(stepwise_model)
```

```
Call:
multinom(formula = prog2 ~ ses + math + science + socst, data = data01)

Coefficients:
         (Intercept)  sesmiddle    seshigh       math    science       socst
general     3.925130 -0.3251216 -0.9441242 -0.1184824 0.08012764 -0.04237059
vocation    8.745662  0.9667470 -0.2018126 -0.1386983 0.03516103 -0.09091674

Std. Errors:
         (Intercept) sesmiddle   seshigh       math    science      socst
general     1.458433 0.4871833 0.5701258 0.03162876 0.02751602 0.02324095
vocation    1.616630 0.5410576 0.6563785 0.03494488 0.02794217 0.02471268
```

```
Residual Deviance: 324.0381
AIC: 348.0381
```

(g) Construct a plot of predicted probabilities from your selected model where the math score varies over the observed range. Other predictors should be set at the most common level or mean value as appropriate. Your plot should be similar to the figure shown on slide 58 (Lecture #4). Comment on the relationship.

In [56]:

```
test_pred <- fitted(test)
test_pred
head(test_pred)
```

A matrix: 200 × 3 of type dbl

	academic	general	vocation
1	0.1482764	0.3382454	0.5134781
2	0.1202017	0.1806283	0.6991700
3	0.4186747	0.2368082	0.3445171
4	0.1726885	0.3508384	0.4764731
5	0.1001231	0.1689374	0.7309395
6	0.3533566	0.2377976	0.4088458
7	0.1562562	0.1973504	0.6463934
8	0.1001231	0.1689374	0.7309395
9	0.2331292	0.2203976	0.5464732
10	0.1699402	0.2025531	0.6275067
11	0.2777727	0.3762066	0.3460207
12	0.2917502	0.2336037	0.4746461
13	0.1071687	0.3082195	0.5846118
14	0.2888779	0.2295357	0.4815864
15	0.1482764	0.3382454	0.5134781
16	0.2777727	0.3762066	0.3460207
17	0.3126251	0.3770895	0.3102855
18	0.3293898	0.2330932	0.4375170
19	0.3293898	0.2330932	0.4375170
20	0.6324598	0.2004342	0.1671060
21	0.1998583	0.2121526	0.5879891
22	0.2888779	0.2295357	0.4815864
23	0.3306609	0.3763962	0.2929429
24	0.2777727	0.3762066	0.3460207
25	0.1726885	0.3508384	0.4764731
26	0.3966333	0.2377208	0.3656458
27	0.3676935	0.3727624	0.2595441
28	0.2888779	0.2295357	0.4815864

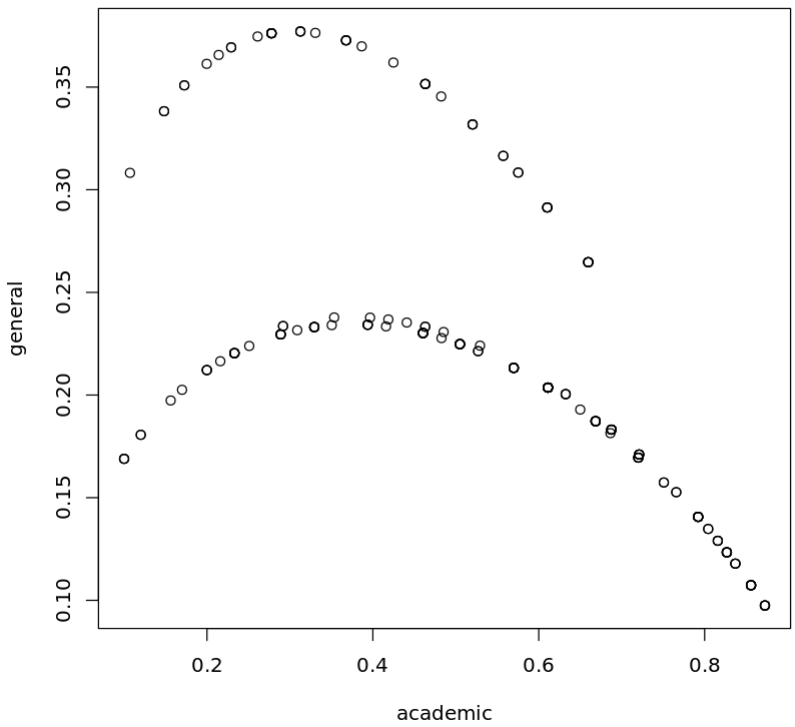
	academic	general	vocation
29	0.2292007	0.3693406	0.4014587
30	0.3865787	0.3698503	0.2435710
⋮	⋮	⋮	⋮
171	0.5572058	0.31650507	0.12628915
172	0.8158376	0.12901647	0.05514591
173	0.6596792	0.26469707	0.07562378
174	0.8726772	0.09748693	0.02983585
175	0.8559136	0.10735910	0.03672735
176	0.6596792	0.26469707	0.07562378
177	0.7508140	0.15740972	0.09177625
178	0.6596792	0.26469707	0.07562378
179	0.6102774	0.29135200	0.09837060
180	0.6876322	0.18315578	0.12921205
181	0.8370332	0.11788753	0.04507931
182	0.6110367	0.20362512	0.18533822
183	0.8559136	0.10735910	0.03672735
184	0.6110367	0.20362512	0.18533822
185	0.8726772	0.09748693	0.02983585
186	0.8726772	0.09748693	0.02983585
187	0.8043127	0.13477968	0.06090766
188	0.8726772	0.09748693	0.02983585
189	0.8267369	0.12338166	0.04988149
190	0.5049270	0.22479322	0.27027983
191	0.8559136	0.10735910	0.03672735
192	0.8267369	0.12338166	0.04988149
193	0.6110367	0.20362512	0.18533822
194	0.8043127	0.13477968	0.06090766
195	0.8370332	0.11788753	0.04507931
196	0.8559136	0.10735910	0.03672735
197	0.6864017	0.18143036	0.13216795
198	0.7508140	0.15740972	0.09177625
199	0.7200349	0.16949969	0.11046543
200	0.6685375	0.18724728	0.14421521

A matrix: 6 × 3 of type dbl

academic	general	vocation

	academic	general	vocation
1	0.1482764	0.3382454	0.5134781
2	0.1202017	0.1806283	0.6991700
3	0.4186747	0.2368082	0.3445171
4	0.1726885	0.3508384	0.4764731
5	0.1001231	0.1689374	0.7309395
6	0.3533566	0.2377976	0.4088458

In [59]:
```
plot(test_pred)
```



In [43]:
```
# continuous predictor variable write within each level of ses
dwrite <- data.frame(ses = rep(c("low", "middle", "high"), each = 41), write = rep(c(30:7
```

In [44]:
```
# store the predicted probabilities for each value of ses and write
pp_write <- cbind(dwrite, predict(test, newdata = dwrite, type = "probs", se = TRUE))
```

In [45]:
```
## calculate the mean probabilities within each level of ses
by(pp_write[, 3:5], pp_write$ses, colMeans)
```

```
pp_write$ses: high
 academic   general  vocation
0.6164315 0.1808037 0.2027648
------------------------------------------------------------
pp_write$ses: low
```

```
 academic   general  vocation 
0.3972977 0.3278174 0.2748849 
------------------------------------------------------------
pp_write$ses: middle
 academic   general  vocation 
0.4256198 0.2010864 0.3732938 
```

In [46]:

```
## melt data set to long for ggplot2
long_pp <- melt(pp_write, id.vars = c("ses", "write"), value.name = "probability")
long_pp
```

A data.frame: 369 × 4

ses	write	variable	value
<chr>	<int>	<fct>	<dbl>
low	30	academic	0.09843588
low	31	academic	0.10716868
low	32	academic	0.11650390
low	33	academic	0.12645834
low	34	academic	0.13704576
low	35	academic	0.14827643
low	36	academic	0.16015670
low	37	academic	0.17268854
low	38	academic	0.18586924
low	39	academic	0.19969106
low	40	academic	0.21414101
low	41	academic	0.22920069
low	42	academic	0.24484625
low	43	academic	0.26104842
low	44	academic	0.27777271
low	45	academic	0.29497963
low	46	academic	0.31262509
low	47	academic	0.33066089
low	48	academic	0.34903525
low	49	academic	0.36769349
low	50	academic	0.38657868
low	51	academic	0.40563241
low	52	academic	0.42479552
low	53	academic	0.44400888
low	54	academic	0.46321407
low	55	academic	0.48235410

ses	write	variable	value
<chr>	<int>	<fct>	<dbl>
low	56	academic	0.50137403
low	57	academic	0.52022153
low	58	academic	0.53884736
low	59	academic	0.55720577
⋮	⋮	⋮	⋮
high	41	vocation	0.30362151
high	42	vocation	0.28400507
high	43	vocation	0.26503065
high	44	vocation	0.24675343
high	45	vocation	0.22921944
high	46	vocation	0.21246521
high	47	vocation	0.19651782
high	48	vocation	0.18139510
high	49	vocation	0.16710605
high	50	vocation	0.15365145
high	51	vocation	0.14102458
high	52	vocation	0.12921205
high	53	vocation	0.11819464
high	54	vocation	0.10794824
high	55	vocation	0.09844474
high	56	vocation	0.08965284
high	57	vocation	0.08153887
high	58	vocation	0.07406758
high	59	vocation	0.06720272
high	60	vocation	0.06090766
high	61	vocation	0.05514591
high	62	vocation	0.04988149
high	63	vocation	0.04507931
high	64	vocation	0.04070545
high	65	vocation	0.03672735
high	66	vocation	0.03311395
high	67	vocation	0.02983585
high	68	vocation	0.02686530
high	69	vocation	0.02417631

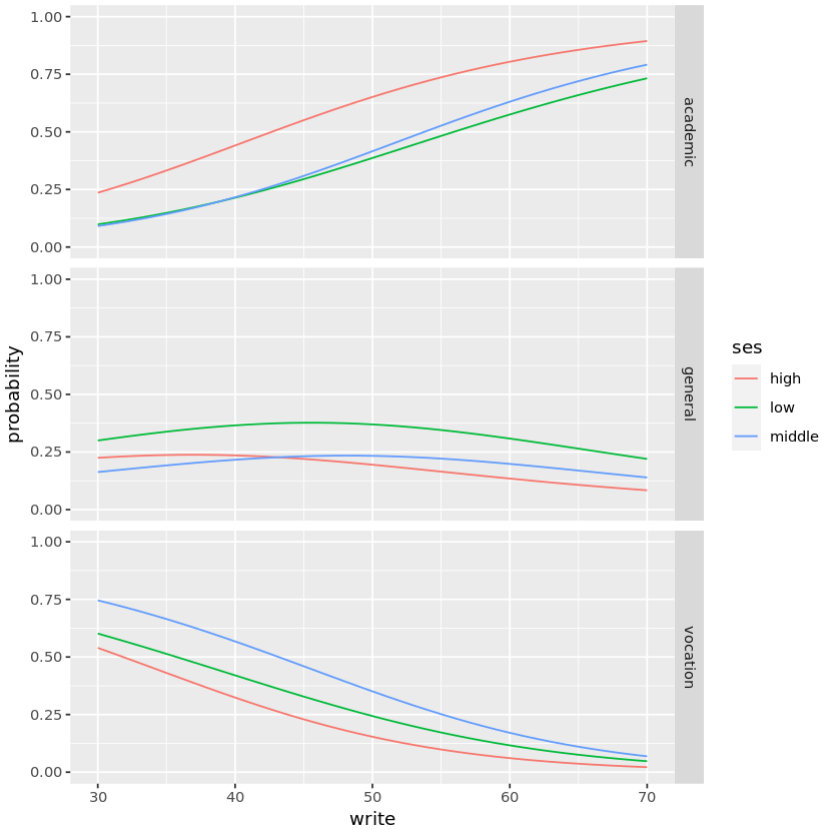
ses	write	variable	value
<chr>	<int>	<fct>	<dbl>
high	70	vocation	0.02174458

In [47]:
```
names(long_pp)[names(long_pp) == "value"] <- "probability"
head(long_pp)
```

A data.frame: 6 × 4

	ses	write	variable	probability
	<chr>	<int>	<fct>	<dbl>
1	low	30	academic	0.09843588
2	low	31	academic	0.10716868
3	low	32	academic	0.11650390
4	low	33	academic	0.12645834
5	low	34	academic	0.13704576
6	low	35	academic	0.14827643

In [48]:
```
## plot predicted probabilities across write values for each level of ses
ggplot(long_pp, aes(x = write, y = probability, colour = ses)) + geom_line() + facet_grid
```



In [ ]:
```

```