MATH 18/19: Overall analysis

George Kinnear

28/11/2020

Read data

Read in the `data_math-1819.csv` file which contains the codes assigned to all items, and produce some summaries of this.

Summary of papers in the sample:

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

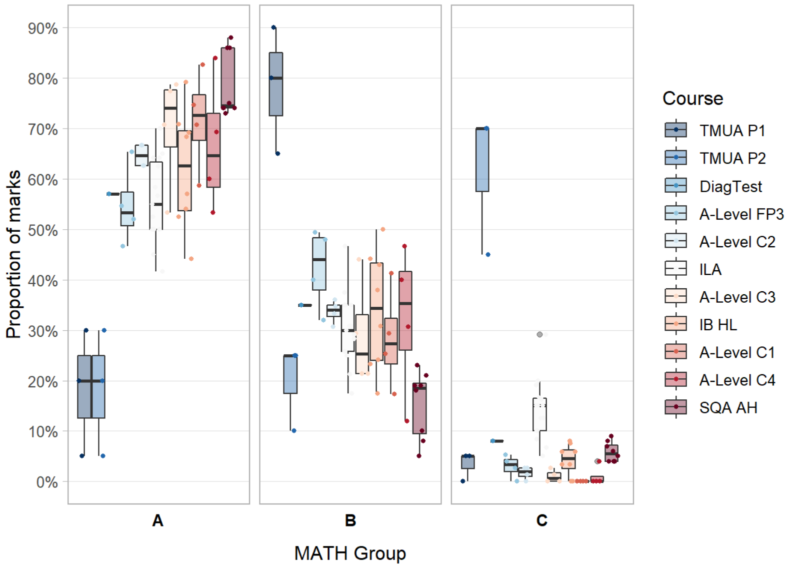
```
## Joining, by = "Course"
```

Course	num_papers	papers	total_question	total_marks
A-Level C1	4	2014, 2015, 2016, 2017	93	300
A-Level C2	4	2014, 2015, 2016, 2017	95	300
A-Level C3	4	2014, 2015, 2016, 2017	95	300
A-Level C4	4	2014, 2015, 2016, 2017	88	300
A-Level FP3	4	2014, 2015, 2016, 2017	78	300
DiagTest	1	2018	24	100
IB HL	4	2014, 2015, 2016, 2017	232	920
ILA	8	2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018	393	1760
SQA AH	8	2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019	244	800
TMUA P1	3	2016, 2017, 2018	60	60
TMUA P2	3	2016, 2017, 2018	60	60

totalqs	totalmarks
1462	5200

Overall proportions

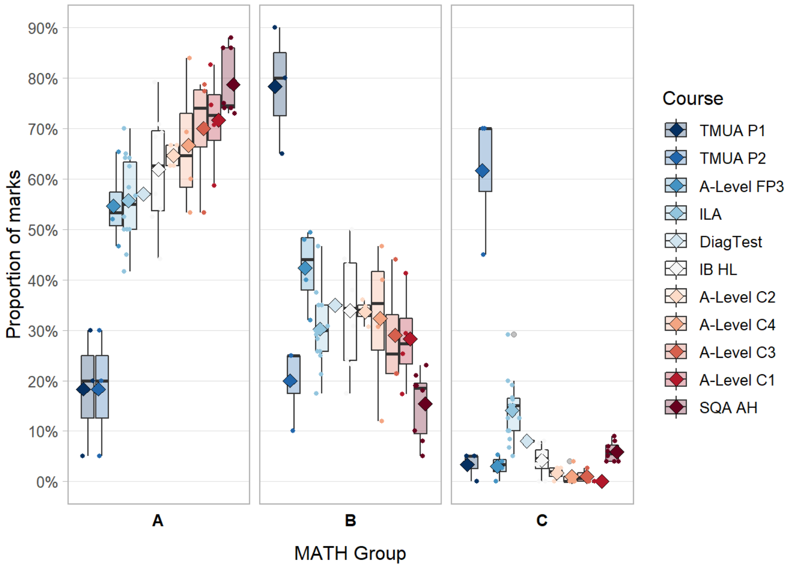
```
MATH_Group_proportions %>%
  # reorder the Course factor by max % group A
  mutate(
    gpAmarks = if_else(MATH_Group=="A",prop_marks,0),
    Course = reorder(Course, gpAmarks, max)
  ) %>%
  ggplot(aes(x=1,y=prop_marks, fill=Course)) +
  geom_boxplot(alpha = 0.4) +
  geom_point(aes(colour=Course),
             position = position_jitterdodge(),
             size = 1) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1),
                     breaks = seq(0,1,0.1)) +
  scale_fill_brewer(palette = "RdBu", direction = -1) +
  scale_colour_brewer(palette = "RdBu", direction = -1) +
  theme_light(base_size = 12)+
  theme(
    panel.grid.minor = element_blank(),
    axis.text.x=element_blank(),
    axis.ticks.x=element_blank(),
    panel.grid.major.x = element_blank(),
    strip.background = element_rect(fill=NA,colour = NA),
    strip.text = element_text(size=10, face="bold", color = "black")
  ) +
  labs(
#    title = "Proportion of marks in each MATH Group",
#    subtitle = "Exams are ordered by proportion of Group A marks",
    x = "MATH Group",
    y = "Proportion of marks"
  ) +
  facet_grid( ~ MATH_Group, switch = "x") +
  ggsave("figs/ABC_allpapers_ranking.pdf",width=15,height=10,units="cm",dpi=300)
```



In this version, we sort by the mean % of group A, and show the means with diamonds

```
MATH_Group_proportions %>%
  # reorder the Course factor by mean % group A
  mutate(
    gpAmarks = if_else(MATH_Group=="A",prop_marks,0),
    Course = reorder(Course, gpAmarks, mean)
  ) %>%
  ggplot(aes(x=1,y=prop_marks, fill=Course)) +
  geom_boxplot(alpha = 0.3) +
  stat_summary(fun.y = "mean",
               geom = "point",
               shape = 23,
               size = 3,
               aes(fill = Course),
               position = position_dodge(0.8)) +
  geom_point(aes(colour=Course),
             position = position_jitterdodge(),
             size = 1) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1),
                     breaks = seq(0,1,0.1)) +
  #  scale_fill_brewer(palette = "Set1") +
  #  scale_colour_brewer(palette = "Set1") +
  scale_fill_brewer(palette = "RdBu", direction = -1) +
  scale_colour_brewer(palette = "RdBu", direction = -1) +
  theme_light(base_size = 12)+
  theme(
    panel.grid.minor = element_blank(),
    axis.text.x=element_blank(),
    axis.ticks.x=element_blank(),
    panel.grid.major.x = element_blank(),
    strip.background = element_rect(fill=NA,colour = NA),
    strip.text = element_text(size=10, face="bold", color = "black")
  ) +
  labs(
    #    title = "Proportion of marks in each MATH Group",
    #    subtitle = "Exams are ordered by proportion of Group A marks",
    x = "MATH Group",
    y = "Proportion of marks"
  ) +
  facet_grid( ~ MATH_Group, switch = "x") +
  ggsave("figs/ABC_allpapers_boxplots.pdf",width=15,height=10,units="cm",dpi=300)
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```

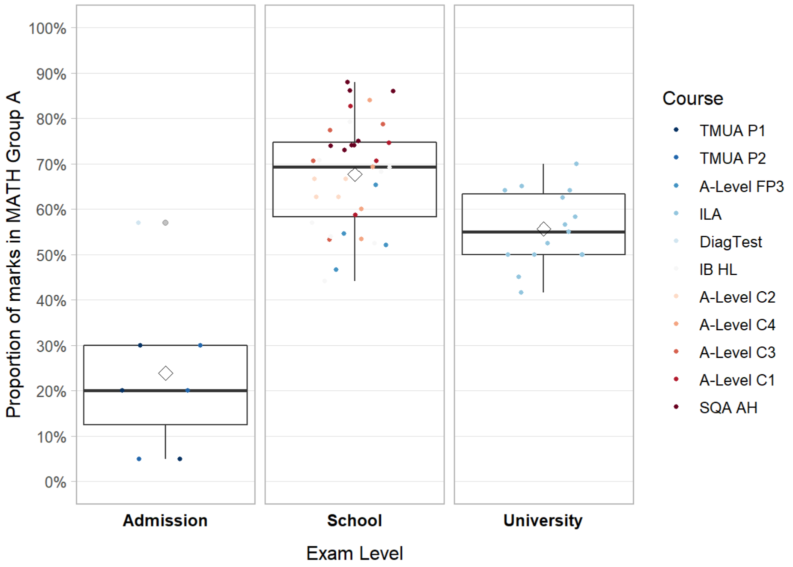


Here we group courses at different education levels

```
gpAschooluni = MATH_Group_proportions %>%
  # reorder the Course factor by mean % group A
  mutate(
    gpAmarks = if_else(MATH_Group=="A",prop_marks,0),
    Course = reorder(Course, gpAmarks, mean),
    CourseGroup = case_when(
      Course %in% c("TMUA P1", "TMUA P2", "DiagTest") ~ "Admission",
      Course == "ILA" ~ "University",
      TRUE ~ "School"
    )
  ) %>%
  filter( MATH_Group == "A" ) %>%
  select(Course, Year, Paper, total_marks_for_Group, total_marks_for_paper, prop_marks, CourseGroup)

gpAschooluni %>%
  ggplot(aes(x=1,y=prop_marks)) +
  geom_boxplot(alpha = 0.3) +
  stat_summary(fun.y = "mean",
               geom = "point",
               shape = 23,
               size = 3) +
  geom_point(aes(colour=Course),
             size = 1,
             position = position_jitter(0.2)) +
  #  geom_text(aes(label=Course)) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1),
                     breaks = seq(0,1,0.1),
                     limits = c(0,1)) +
  #  scale_fill_brewer(palette = "Set1") +
  #  scale_colour_brewer(palette = "Set1") +
  scale_fill_brewer(palette = "RdBu", direction = -1) +
  scale_colour_brewer(palette = "RdBu", direction = -1) +
  theme_light(base_size = 12)+
  theme(
    panel.grid.minor = element_blank(),
    axis.text.x=element_blank(),
    axis.ticks.x=element_blank(),
    panel.grid.major.x = element_blank(),
    strip.background = element_rect(fill=NA,colour = NA),
    strip.text = element_text(size=10, face="bold", color = "black")
  ) +
  labs(
    #    title = "Proportion of marks in each MATH Group",
    #    subtitle = "Exams are ordered by proportion of Group A marks",
    x = "Exam Level",
    y = "Proportion of marks in MATH Group A"
  ) +
  facet_grid( ~ CourseGroup, switch = "x") +
  ggsave("figs/A_mean_school_vs_uni.pdf",width=15,height=10,units="cm",dpi=300)
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```



Statistical comparisons

t-test comparison by level

```
gpAschool = gpAschooluni %>% filter(CourseGroup=="School") %>% select(prop_marks) %>% data.matrix
gpA_Alevel = gpAschooluni %>% filter(str_detect(Course, "A-Level")) %>% select(prop_marks) %>% data.matrix
gpAuni = gpAschooluni %>% filter(Course=="ILA") %>% select(prop_marks) %>% data.matrix

t.test(gpAschool, gpAuni)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  gpAschool and gpAuni
## t = 4.1755, df = 37.021, p-value = 0.0001733
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.06174592 0.17816149
## sample estimates:
## mean of x mean of y 
## 0.6766204 0.5566667
```

```
t.test(gpA_Alevel, gpAuni)
```

```
## 
##  Welch Two Sample t-test
## 
## data:  gpA_Alevel and gpAuni
## t = 3.0996, df = 32.957, p-value = 0.00395
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.03390045 0.16343289
## sample estimates:
## mean of x mean of y 
## 0.6553333 0.5566667
```

Bayesian comparison

```
BESTout = BESTmcmc(gpAschool,gpAuni)
```

```
## Waiting for parallel processing to complete...done.
```

```
BESTout
```

```
## MCMC fit results for BEST analysis:
## 100002 simulations saved.
##            mean       sd   median   HDIlo    HDIup Rhat n.eff
## mu1     0.67727  0.02060  0.67726 0.63731   0.7185    1 58029
## mu2     0.55629  0.02424  0.55636 0.50903   0.6048    1 58749
## nu     41.76518 31.40481 33.26935 3.47672 103.9886    1 22769
## sigma1  0.11824  0.01558  0.11681 0.08931   0.1493    1 47375
## sigma2  0.08872  0.01976  0.08571 0.05540   0.1287    1 36287
## 
## 'HDIlo' and 'HDIup' are the limits of a 95% HDI credible interval.
## 'Rhat' is the potential scale reduction factor (at convergence, Rhat=1).
## 'n.eff' is a crude measure of effective sample size.
```

```
summary(BESTout)
```

```
##               mean  median    mode HDI%   HDIlo    HDIup compVal %>compVal
## mu1         0.6773  0.6773  0.6762   95  0.6373   0.7185                  
## mu2         0.5563  0.5564  0.5568   95  0.5090   0.6048                  
## muDiff      0.1210  0.1211  0.1211   95  0.0583   0.1840       0     100.0
## sigma1      0.1182  0.1168  0.1143   95  0.0893   0.1493                  
## sigma2      0.0887  0.0857  0.0801   95  0.0554   0.1287                  
## sigmaDiff   0.0295  0.0308  0.0324   95 -0.0211   0.0784       0      88.7
## nu         41.7652 33.2693 17.0551   95  3.4767 103.9886                  
## log10nu     1.5070  1.5220  1.5716   95  0.8631   2.1137                  
## effSz       1.1656  1.1645  1.1674   95  0.5181   1.8192       0     100.0
```

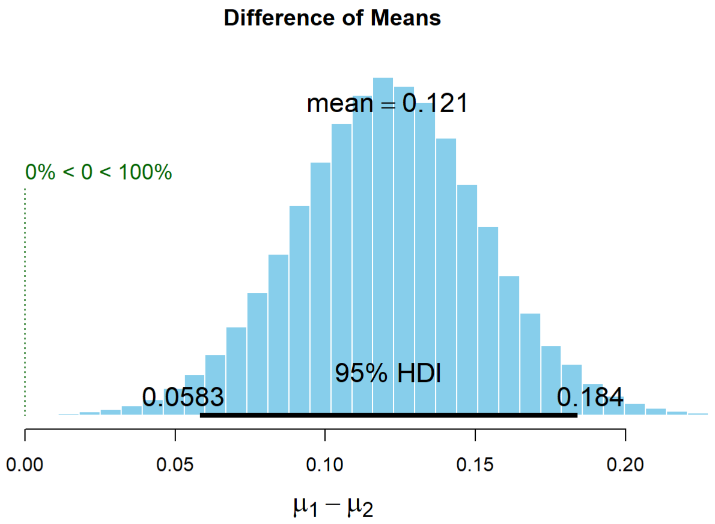
This gives the mean proportion of group A:

- for school exams, 0.68 (95% HDI [0.64, 0.72])
- for university exams, 0.56 (95% HDI [0.51, 0.60])

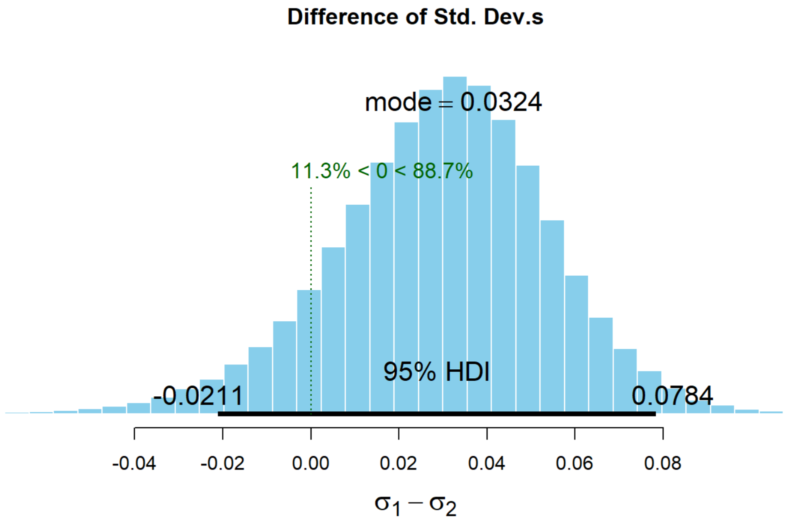
The difference between these is 0.12 (95% HDI [0.06, 0.18]).

Here are some more diagostic plots for this summary:

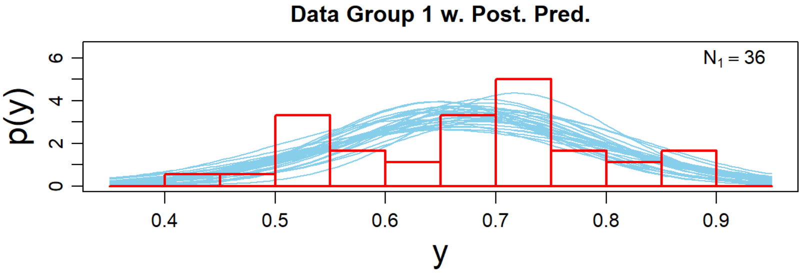
```
plot(BESTout)
```

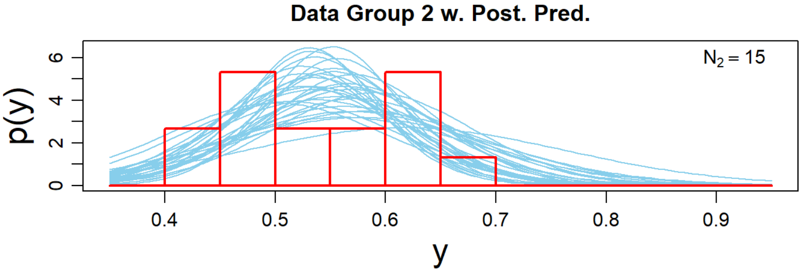


```
plot(BESTout, "sd")
```

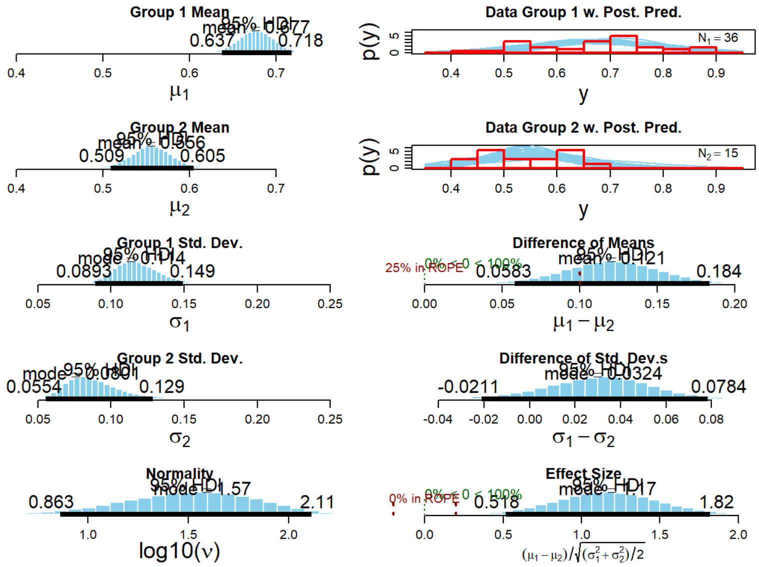


```
plotPostPred(BESTout)
```

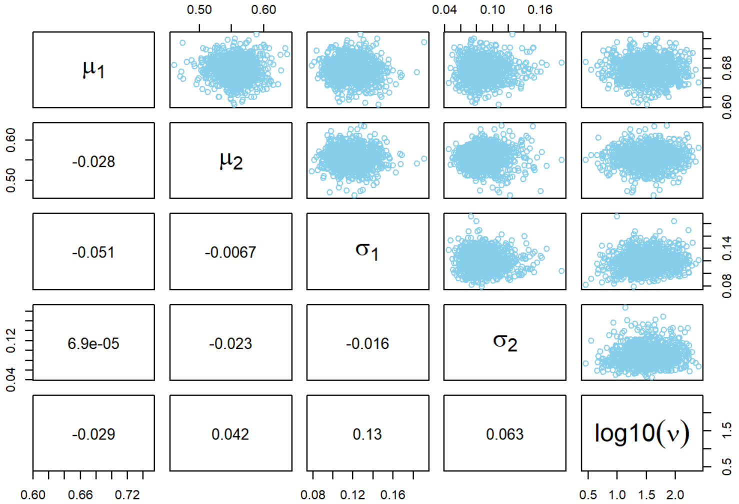




```
plotAll(BESTout, credMass=0.95, ROPEm=c(-0.1,0.1),
        ROPEeff=c(-0.2,0.2), compValm=0.5)
```



```
pairs(BESTout)
```



Looking at university versus A-Level in particular:

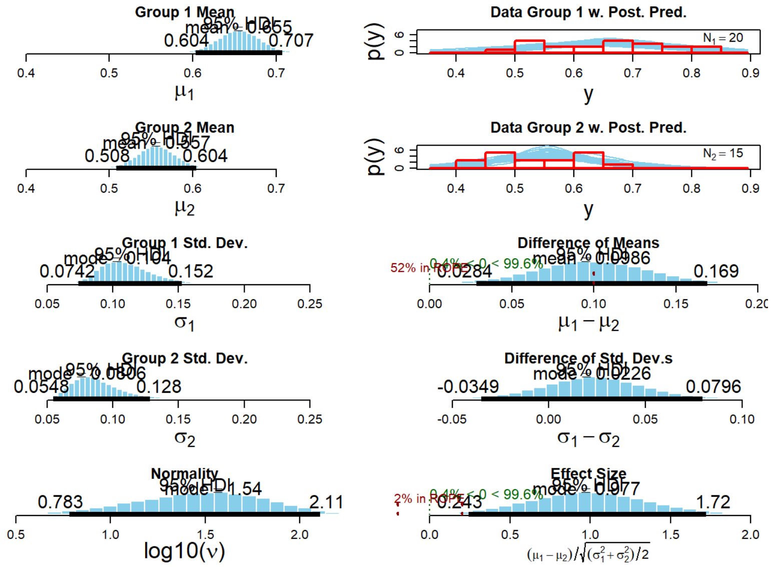
```
BESTout_Alevel = BESTmcmc(gpA_Alevel,gpAuni)
```

```
## Waiting for parallel processing to complete...done.
```

```
summary(BESTout_Alevel)
```

```
##               mean  median    mode HDI%   HDIlo    HDIup compVal %>compVal
## mu1         0.6551  0.6551  0.6572   95  0.6035   0.7073                  
## mu2         0.5565  0.5563  0.5547   95  0.5085   0.6043                  
## muDiff      0.0986  0.0985  0.0942   95  0.0284   0.1691       0      99.6
## sigma1      0.1114  0.1087  0.1035   95  0.0742   0.1524                  
## sigma2      0.0885  0.0854  0.0806   95  0.0548   0.1283                  
## sigmaDiff   0.0229  0.0229  0.0226   95 -0.0349   0.0796       0      80.6
## nu         39.2057 30.7896 14.1434   95  2.3815 100.3458                  
## log10nu     1.4673  1.4884  1.5428   95  0.7833   2.1080                  
## effSz       0.9919  0.9883  0.9773   95  0.2426   1.7235       0      99.6
```

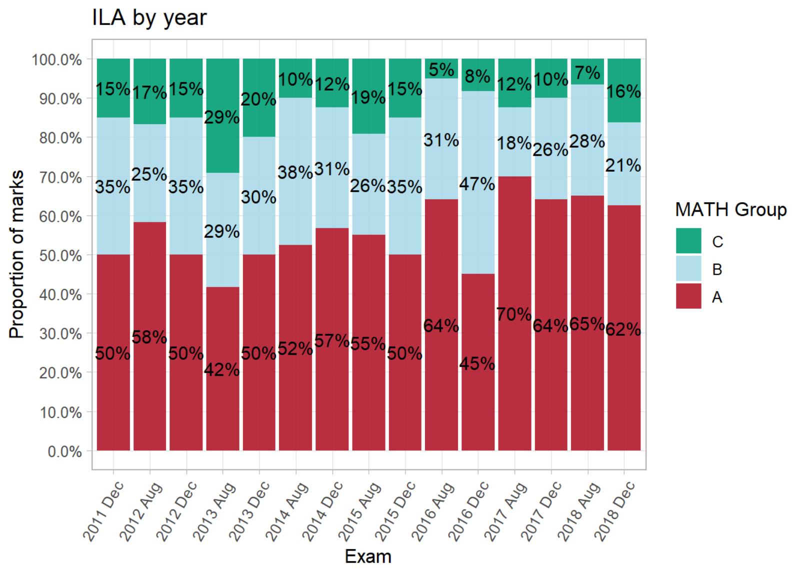
```
plotAll(BESTout_Alevel, credMass=0.95, ROPEm=c(-0.1,0.1),
        ROPEeff=c(-0.2,0.2), compValm=0.5)
```



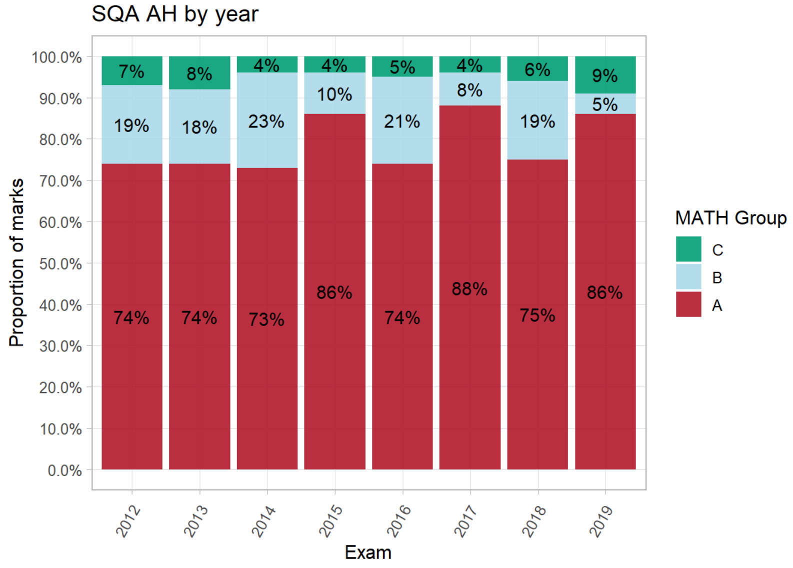
Details of each paper

Stacked bars of ABC proprtion by exam paper

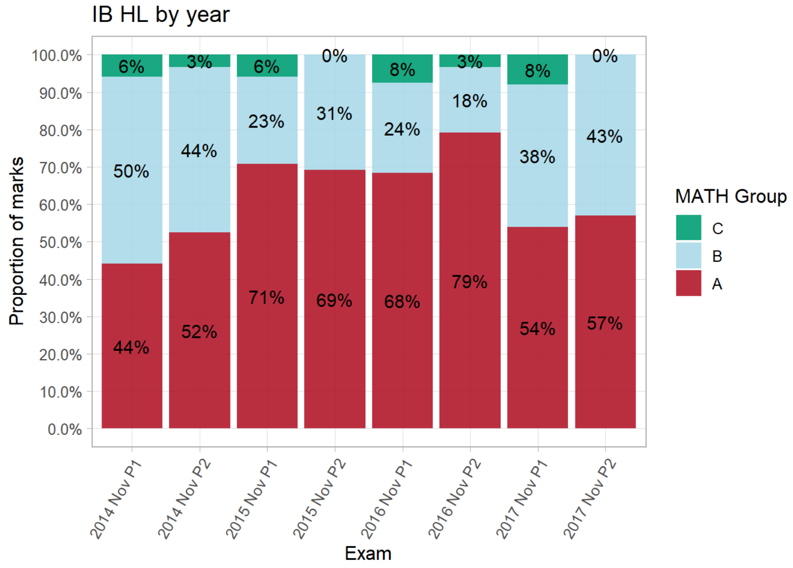
```
math_stacked_bars <- function(course_name) {
  MATH_Group_proportions %>%
    filter(Course == course_name) %>%
    # compute the positions of the text labels
    # see http://t-redactyl.io/blog/2016/01/creating-plots-in-r-using-ggplot2-part-4-stacked-bar-plots.html
    ddply(.(Course,Exam), transform, pos = cumsum(prop_marks) - (0.5 * prop_marks)) %>%
    ggplot(aes(x = Exam,
               y = prop_marks,
               fill = fct_rev(MATH_Group))) +
    geom_bar(stat="identity", alpha = 0.9) +
    geom_text(aes(x = Exam, y = 1-pos, label = paste0(sprintf("%.0f",100*prop_marks),"%")),
              size=4) +
    scale_y_continuous(labels = scales::percent,
                       breaks = seq(0,1,0.1)) +
    scale_fill_manual(values = palMATH) +
    theme_light(base_size = 12)+
    theme(
      panel.grid.minor = element_blank(),
      axis.text.x = element_text(angle = 60, hjust = 1)
    ) +
    labs(
      title = paste(course_name, "by year"),
      fill = "MATH Group",
      x = "Exam",
      y = "Proportion of marks"
    )
}
math_stacked_bars("ILA")
```



```
math_stacked_bars("SQA AH")
```



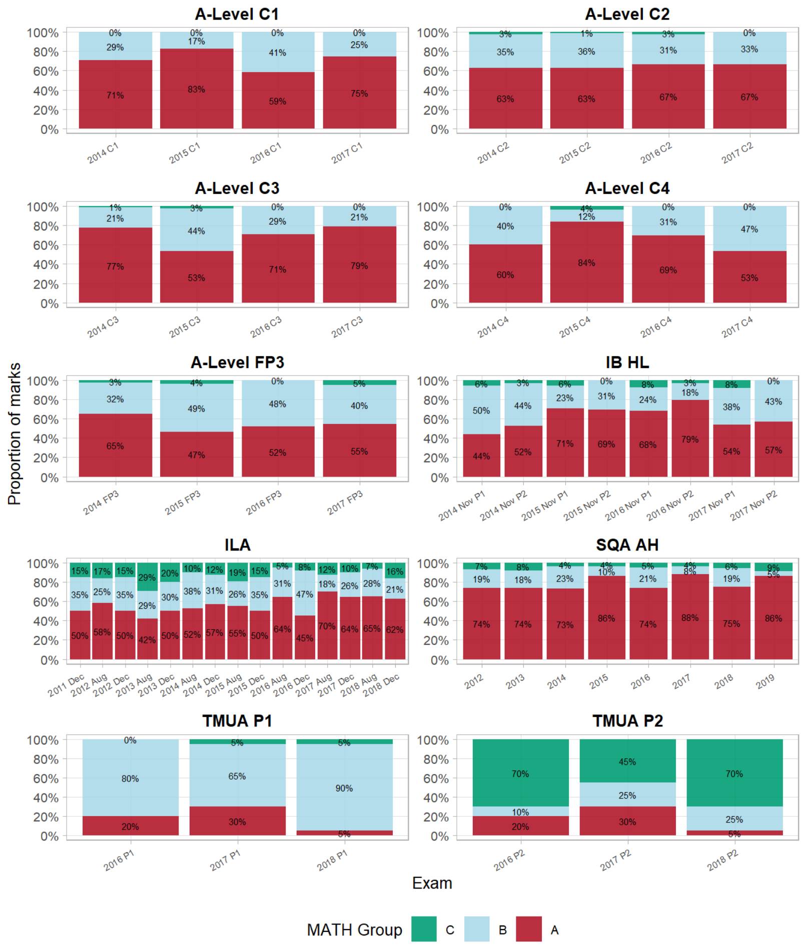
```
math_stacked_bars("IB HL")
```



```
#math_stacked_bars("A-Level C1-C4")
#math_stacked_bars("A-Level FP3")
```

Do them all in one big plot, using facets:

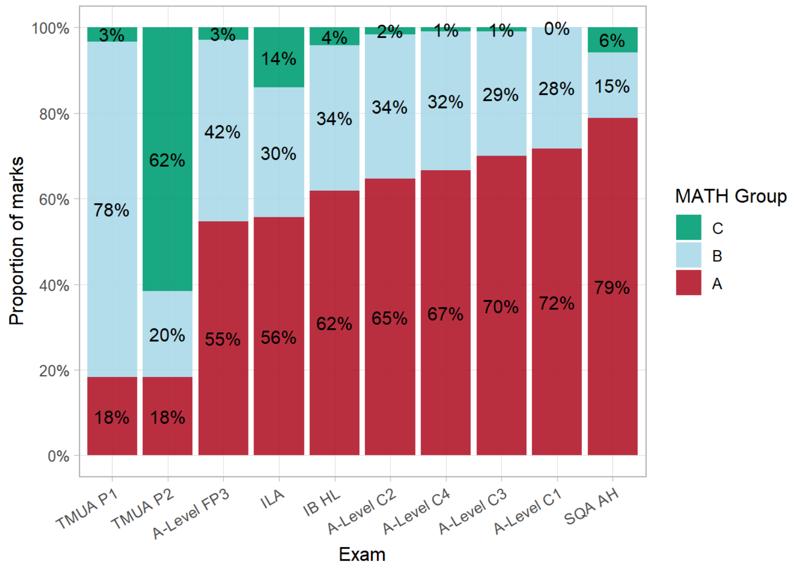
```
MATH_Group_proportions %>%
  filter(!Course %in% c("DiagTest", "TMUA")) %>%
  ddply(.(Course,Exam), transform, pos = cumsum(prop_marks) - (0.5 * prop_marks)) %>%
  ggplot(aes(x = Exam,
             y = prop_marks,
             fill = fct_rev(MATH_Group))) +
  geom_bar(stat="identity", alpha = 0.9) +
  geom_text(aes(x = Exam, y = 1-pos, label = paste0(sprintf("%.0f",100*prop_marks),"%")),
            size=2) +
  scale_y_continuous(labels = scales::percent,
                     breaks = seq(0,1,0.2)) +
  scale_fill_manual(values = palMATH) +
  theme_light(base_size = 10)+
  theme(
    panel.grid.minor = element_blank(),
    axis.text.x = element_text(size = rel(0.7), angle = 30, hjust = 1),
    strip.background = element_rect(fill=NA,colour = NA),
    strip.text = element_text(size=10, face="bold", color = "black"),
    legend.position="bottom"
  ) +
  labs(
    fill = "MATH Group",
    x = "Exam",
    y = "Proportion of marks"
  ) +
  facet_wrap( ~ Course, ncol = 2, scales = "free") +
  ggsave("figs/ABC_allpapers_facets.pdf",width=15,height=20,units="cm",dpi=300)
```



A version of Darlington 2015 Fig 2/3:

```
MATH_Group_proportions %>%
  filter(!Course %in% c("TMUA", "DiagTest")) %>%
  ungroup() %>%
  group_by(Course,MATH_Group) %>%
  summarise(
    prop_marks = mean(prop_marks),
    n = n()
  ) %>%
  ungroup() %>%
  mutate(
    gpAmarks = if_else(MATH_Group=="A",prop_marks,0),
#    Course = paste(Course,"\n(n=",n,")"),
    Course = (reorder(Course, gpAmarks, max))
  ) %>%
  ddply(.(Course), transform, pos = cumsum(prop_marks) - (0.5 * prop_marks)) %>%
  ggplot(aes(x = Course,
             y = prop_marks,
             fill = fct_rev(MATH_Group))) +
  geom_bar(stat="identity", alpha = 0.9) +
  geom_text(aes(x = Course, y = pos, label = paste0(sprintf("%.0f",100*prop_marks),"%")),
            size=4) +
  scale_y_continuous(labels = scales::percent,
                     breaks = seq(0,1,0.2)) +
  scale_fill_manual(values = palMATH) +
  theme_light(base_size = 12)+
  theme(
    panel.grid.minor = element_blank(),
    axis.text.x = element_text(angle = 30, hjust = 1)
  ) +
  labs(
#    title = "Mean proportion of marks in each MATH Group",
    fill = "MATH Group",
    x = "Exam",
    y = "Proportion of marks"
  ) +
  ggsave("figs/ABC_school_and_ILA_means.pdf",width=15,height=10,units="cm",dpi=300)
```

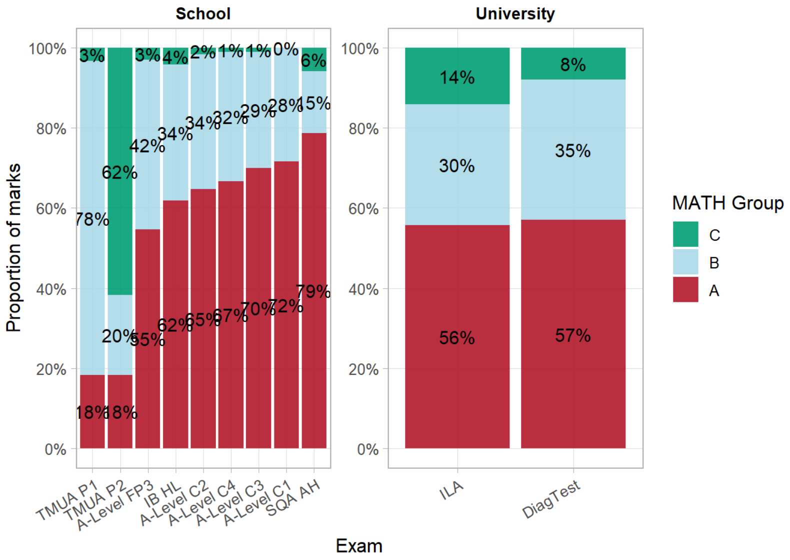
```
## `summarise()` regrouping output by 'Course' (override with `.groups` argument)
```



Try using facets to show school vs university:

```
MATH_Group_proportions %>%
  ungroup() %>%
  group_by(Course,MATH_Group) %>%
  summarise(
    prop_marks = mean(prop_marks),
    n = n()
  ) %>%
  ungroup() %>%
  mutate(
    gpAmarks = if_else(MATH_Group=="A",prop_marks,0),
    #    Course = paste(Course,"\n(n=",n,")"),
    Course = (reorder(Course, gpAmarks, max)),
    CourseGroup = if_else(Course %in% c("TMUA", "DiagTest", "ILA"), "University", "School")
  ) %>%
  ddply(.(Course), transform, pos = cumsum(prop_marks) - (0.5 * prop_marks)) %>%
  ggplot(aes(x = Course,
             y = prop_marks,
             fill = fct_rev(MATH_Group))) +
  geom_bar(stat="identity", alpha = 0.9) +
  geom_text(aes(x = Course, y = pos, label = paste0(sprintf("%.0f",100*prop_marks),"%")),
            size=4) +
  scale_y_continuous(labels = scales::percent,
                     breaks = seq(0,1,0.2)) +
  scale_fill_manual(values = palMATH) +
  facet_wrap( ~ CourseGroup, ncol = 2, scales = "free") +
  theme_light(base_size = 12)+
  theme(
    panel.grid.minor = element_blank(),
    axis.text.x = element_text(angle = 30, hjust = 1),
    strip.background = element_rect(fill=NA,colour = NA),
    strip.text = element_text(size=10, face="bold", color = "black")
  ) +
  labs(
    #    title = "Mean proportion of marks in each MATH Group",
    fill = "MATH Group",
    x = "Exam",
    y = "Proportion of marks"
  ) +
  ggsave("figs/ABC_school_vs_other_means.pdf",width=15,height=10,units="cm",dpi=300)
```

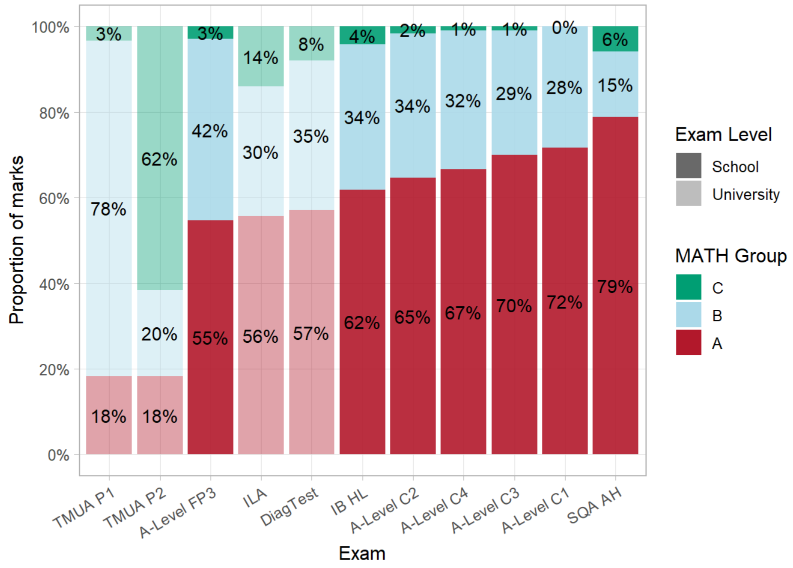
```
## `summarise()` regrouping output by 'Course' (override with `.groups` argument)
```



Showing the school/uni distinction using opacity:

```
MATH_Group_proportions %>%
  ungroup() %>%
  group_by(Course,MATH_Group) %>%
  summarise(
    prop_marks = mean(prop_marks),
    n = n()
  ) %>%
  ungroup() %>%
  mutate(
    gpAmarks = if_else(MATH_Group=="A",prop_marks,0),
    #    Course = paste(Course,"\n(n=",n,")"),
    Course = (reorder(Course, gpAmarks, max)),
    CourseGroup = if_else(Course %in% c("TMUA P1", "TMUA P2", "DiagTest", "ILA"), "University", "School")
  ) %>%
  ddply(.(Course), transform, pos = cumsum(prop_marks) - (0.5 * prop_marks)) %>%
  ggplot(aes(x = Course,
             y = prop_marks,
             fill = fct_rev(MATH_Group))) +
  geom_bar(stat="identity",
           aes(alpha = CourseGroup)) +
  geom_text(aes(x = Course, y = pos, label = paste0(sprintf("%.0f",100*prop_marks),"%")),
            size=4) +
  scale_y_continuous(labels = scales::percent,
                     breaks = seq(0,1,0.2)) +
  scale_fill_manual(values = palMATH) +
  scale_alpha_manual("Exam Level", values = c("School" = 0.9, "University" = 0.4)) +
  theme_light(base_size = 12)+
  theme(
    panel.grid.minor = element_blank(),
    axis.text.x = element_text(angle = 30, hjust = 1)
  ) +
  labs(
    #    title = "Mean proportion of marks in each MATH Group",
    fill = "MATH Group",
    x = "Exam",
    y = "Proportion of marks"
  ) +
  ggsave("figs/ABC_all_means.pdf",width=15,height=10,units="cm",dpi=300)
```

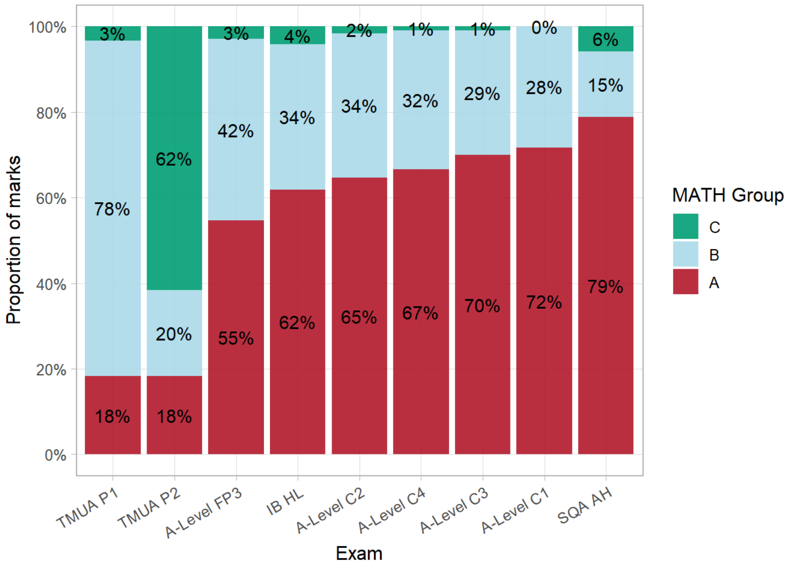
```
## `summarise()` regrouping output by 'Course' (override with `.groups` argument)
```



Focusing on school exams:

```
MATH_Group_proportions %>%
  filter(!Course %in% c("TMUA", "DiagTest", "ILA")) %>%
  ungroup() %>%
  group_by(Course,MATH_Group) %>%
  summarise(
    prop_marks = mean(prop_marks),
    n = n()
  ) %>%
  ungroup() %>%
  mutate(
    gpAmarks = if_else(MATH_Group=="A",prop_marks,0),
    #    Course = paste(Course,"\n(n=",n,")"),
    Course = (reorder(Course, gpAmarks, max))
  ) %>%
  ddply(.(Course), transform, pos = cumsum(prop_marks) - (0.5 * prop_marks)) %>%
  ggplot(aes(x = Course,
             y = prop_marks,
             fill = fct_rev(MATH_Group))) +
  geom_bar(stat="identity", alpha = 0.9) +
  geom_text(aes(x = Course, y = pos, label = paste0(sprintf("%.0f",100*prop_marks),"%")),
            size=4) +
  scale_y_continuous(labels = scales::percent,
                     breaks = seq(0,1,0.2)) +
  scale_fill_manual(values = palMATH) +
  theme_light(base_size = 12)+
  theme(
    panel.grid.minor = element_blank(),
    axis.text.x = element_text(angle = 30, hjust = 1)
  ) +
  labs(
    #    title = "Mean proportion of marks in each MATH Group",
    fill = "MATH Group",
    x = "Exam",
    y = "Proportion of marks"
  ) +
  ggsave("figs/ABC_school_means.pdf",width=15,height=10,units="cm",dpi=300)
```

```
## `summarise()` regrouping output by 'Course' (override with `.groups` argument)
```



Detailed MATH categories

Like Darlington (2014) Fig 1 - looking in detail at the MATH categories.

```
MATH_Cat_proportions = ratings %>%
  group_by(Course,Year,Paper,MATH_Group,MATH) %>%
  summarise(
    total_marks_for_Cat = sum(Marks)
  ) %>%
  ungroup() %>%
  # add in the missing "0%" entries
  complete(nesting(Course,Year,Paper),MATH, fill = list(total_marks_for_Cat = 0)) %>%
  mutate(
    # restore the MATH_Group for the "0%" entries
    MATH_Group = str_sub(MATH,1,1)
  ) %>%
  left_join(MATH_Group_proportions) %>%
  mutate (
    prop_marks_cat = total_marks_for_Cat / total_marks_for_Group
  ) %>%
  ungroup()
```

```
## `summarise()` regrouping output by 'Course', 'Year', 'Paper', 'MATH_Group' (override with `.groups` argument)
```

```
## Joining, by = c("Course", "Year", "Paper", "MATH_Group")
```

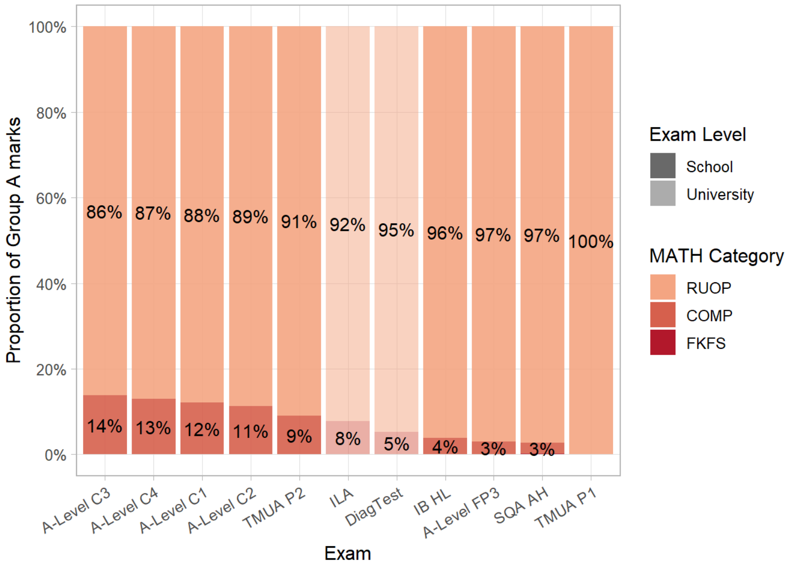
```
MATH_Cat_props_by_Course = MATH_Cat_proportions %>%
  group_by(Course, MATH) %>%
  summarise(
    total_Cat = sum(total_marks_for_Cat),
    total_Group = sum(total_marks_for_Group),
    prop_of_Group = total_Cat / total_Group * 100
  )
```

```
## `summarise()` regrouping output by 'Course' (override with `.groups` argument)
```

```
MATH_Cat_props_by_Course %>%
  select(-total_Cat,-total_Group) %>%
  spread(MATH, prop_of_Group) %>%
  knitr::kable(booktabs = T, digits = 0)
```

Course	A1	A2	A3	B1	B2	C1	C2	C3
A-Level C1	0	12	88	48	52	NaN	NaN	NaN
A-Level C2	0	11	89	30	70	100	0	0
A-Level C3	0	14	86	31	69	67	33	0
A-Level C4	0	13	87	19	81	100	0	0
A-Level FP3	0	3	97	11	89	100	0	0
DiagTest	0	5	95	86	14	0	100	0
IB HL	0	4	96	50	50	97	0	3
ILA	0	8	92	44	56	43	57	0
SQA AH	0	3	97	61	39	91	9	0
TMUA P1	0	0	100	43	57	50	50	0
TMUA P2	0	9	91	58	42	73	27	0

```
MATH_Cat_props_by_Course %>%
  filter( str_sub(MATH,1,1) == "A") %>%
  ungroup() %>%
  mutate(
    Course = (reorder(Course, prop_of_Group, max)),
    CourseGroup = if_else(Course %in% c("TMUA", "DiagTest", "ILA"), "University", "School"),
    MATH = fct_recode(MATH, FKFS = "A1", COMP = "A2", RUOP = "A3"),
    prop_of_Group = prop_of_Group / 100
  ) %>%
  ddply(.(Course), transform, pos = cumsum(prop_of_Group) - (0.5 * prop_of_Group)) %>%
  ggplot(aes(x = Course,
             y = prop_of_Group,
             fill = fct_rev(MATH),
             alpha = CourseGroup)) +
  geom_bar(stat="identity") +
  geom_text(aes(x = Course,
                y = pos,
                label = if_else(prop_of_Group <0.01,
                                "",
                                paste0(sprintf("%.0f",100*prop_of_Group),"%"))
                ),
            size = 4,
            alpha = 1) +
  scale_alpha_manual("Exam Level", values = c("School" = 0.9, "University" = 0.5)) +
  scale_y_continuous(labels = scales::percent,
                     breaks = seq(0,1,0.2)) +
  scale_fill_manual(values = rev(c(palMATH[3],"#d6604d","#f4a582"))) +
  theme_light(base_size = 12)+
  theme(
    panel.grid.minor = element_blank(),
    axis.text.x = element_text(angle = 30, hjust = 1)
  ) +
  labs(
    #    title = "Mean proportion of marks in each MATH Group",
    fill = "MATH Category",
    x = "Exam",
    y = "Proportion of Group A marks"
  ) +
  ggsave("figs/A_breakdown.pdf",width=15,height=10,units="cm",dpi=300)
```


Proportion of Group A marks
100%
80%
60%
40%
20%
0%
86%
87%
88%
89%
91%
92%
95%
96%
97%
97%
100%
14%
13%
12%
11%
9%
8%
5%
4%
3%
3%
A-Level C3
A-Level C4
A-Level C1
A-Level C2
TMUA P2
ILA
DiagTest
IB HL
A-Level FP3
SQA AH
TMUA P1
Exam
Exam Level
School
University
MATH Category
RUOP
COMP
FKFS