

Scoring Evaluations

Dave Bridges

1/31/2018

Contents

Data Import	1
Assessment Choices	2
Among Students with an A	11
Likelyhood of an A Grade	16
Session Information	17

Data Import

Exported **Assignment Type Summaries** from GradeCraft and imported.

```
library(readr)
library(dplyr)
library(tidyr)
datafile <- 'Principles of Nutritional Sciences Assignment Type Summary - 2018-02-01.csv'

library(readr)
dataset <- read_csv(datafile)

dropped.students <- c('dave.bridges','ajian','zhongyli')

assessment.dataset <-
  dataset %>%
  select(-`First Name`, -`Last Name`, -Email, -Team) %>%
  gather(Assignment, Points, -Username) %>%
  filter(!(Username %in% dropped.students)) %>%
  arrange(Points)

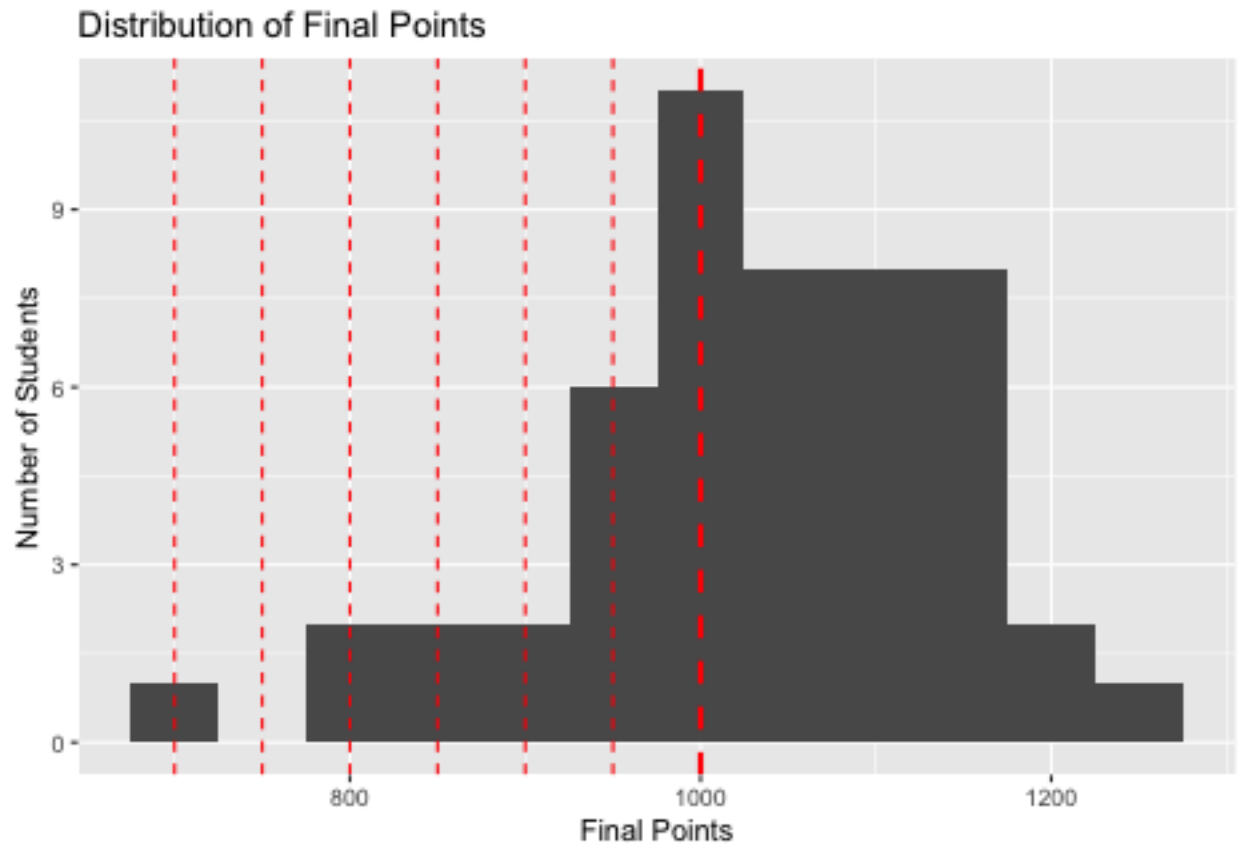
summary.dataset <-
  assessment.dataset %>%
  group_by(Username) %>%
  summarize(Total = sum(Points)) %>%
  arrange(-Total)
```

This script imports the data from **Principles of Nutritional Sciences Assignment Type Summary - 2018-02-01.csv**. This script is located at /Users/davebrid/Documents/GitHub/TeachingLectures/Michigan/NUTR630/Evaluation/Summary/GradeCraft Metrics and was most recently run on Fri Aug 31 11:33:05 2018

```
library(ggplot2)

p <- ggplot(data=summary.dataset, aes(x=Total))
p + geom_histogram(binwidth=50) +
  labs(x="Final Points",y="Number of Students", title="Distribution of Final Points") +
```

```
geom_vline(aes(xintercept=1000), color="red", linetype="dashed", size=1) +
geom_vline(aes(xintercept=950), color="red", linetype="dashed", size=0.5) +
geom_vline(aes(xintercept=900), color="red", linetype="dashed", size=0.5) +
geom_vline(aes(xintercept=850), color="red", linetype="dashed", size=0.5) +
geom_vline(aes(xintercept=800), color="red", linetype="dashed", size=0.5) +
geom_vline(aes(xintercept=750), color="red", linetype="dashed", size=0.5) +
geom_vline(aes(xintercept=700), color="red", linetype="dashed", size=0.5)
```



There were 31 students who earned an A. Among those who earned an A, their average points were 1096.419. This means that the A students averaged **96.419** points too many than they needed. This is relative to the overall mean of 1028.255

Assessment Choices

Aggregated

```
required.assignments <- c('Midterm', 'Class Based Assignments', 'In Class Quiz')

assignment.summary.data <-
  assessment.dataset %>%
  group_by(Assignment) %>%
  summarize(Mean.Points = mean(Points),
            SD.Points = sd(Points)) %>%
  mutate(Required = ifelse(Assignment %in% required.assignments, "Required", "Optional")) %>%
```

```

mutate(Required = relevel(as.factor(Required), ref="Required")) %>%
arrange(Required, -Mean.Points)

assignment.summary.data.required <-
  assessment.dataset %>%
  mutate(Required = ifelse(Assignment %in% required.assignments, "Required", "Optional")) %>%
  mutate(Required = relevel(as.factor(Required), ref="Required")) %>%
  group_by(Username, Required) %>%
  summarize(Points = sum(Points)) %>%
  group_by(Required) %>%
  summarize(Median.Points = median(Points),
            Median.Points.SD = sd(Points)) %>%
  mutate(Relative.Points = Median.Points/sum(Median.Points) *100,
         Relative.Points.SD = Median.Points.SD/sum(Median.Points)*100)

assignment.summary.data.relative <-
  assessment.dataset %>%
  group_by(Username) %>%
  mutate(Total = sum(Points)) %>%
  group_by(Assignment) %>%
  summarize(Relative.Points = mean(Points/Total)*100,
            SD.Relative.Points = sd(Points/Total)*100) %>%
  mutate(Required = ifelse(Assignment %in% required.assignments, "Required", "Optional")) %>%
  mutate(Required = relevel(as.factor(Required), ref="Required")) %>%
  arrange(Required, -Relative.Points)

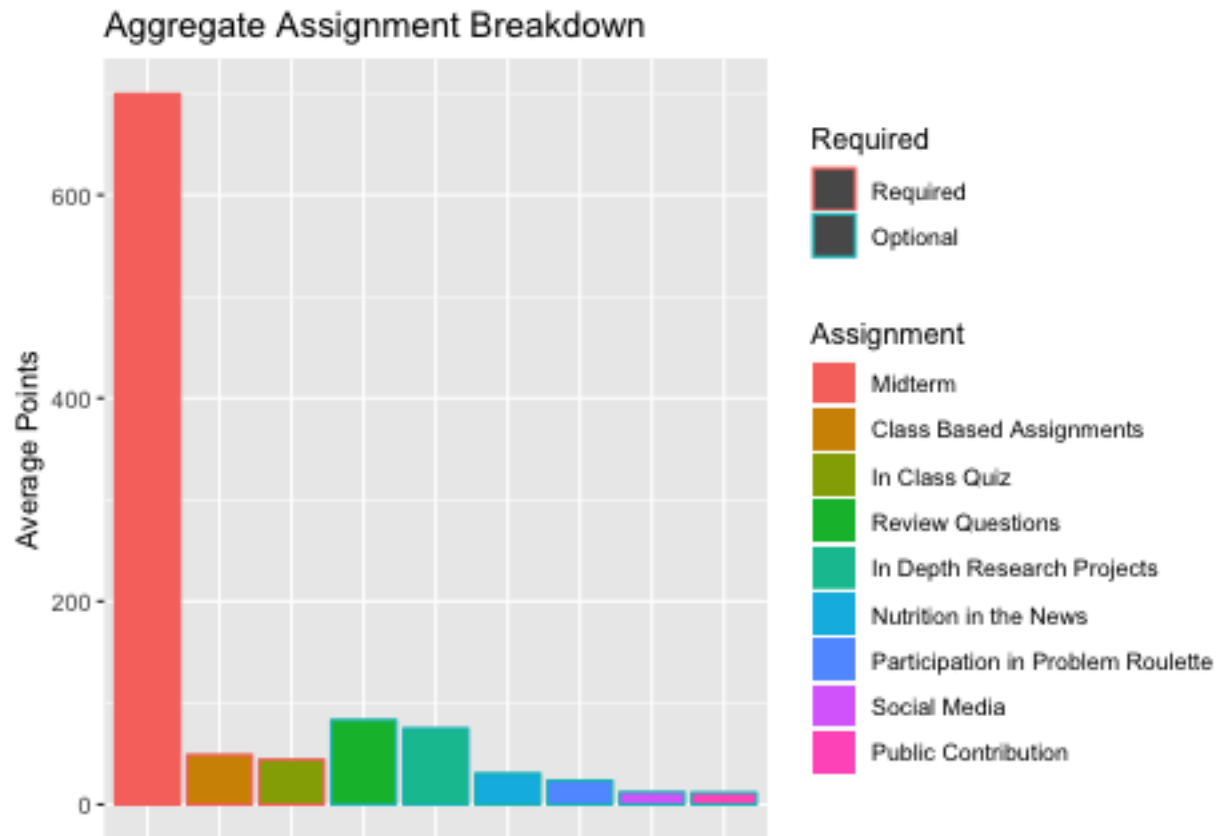
positions <- as.character(assignment.summary.data$Assignment)

library(forcats)

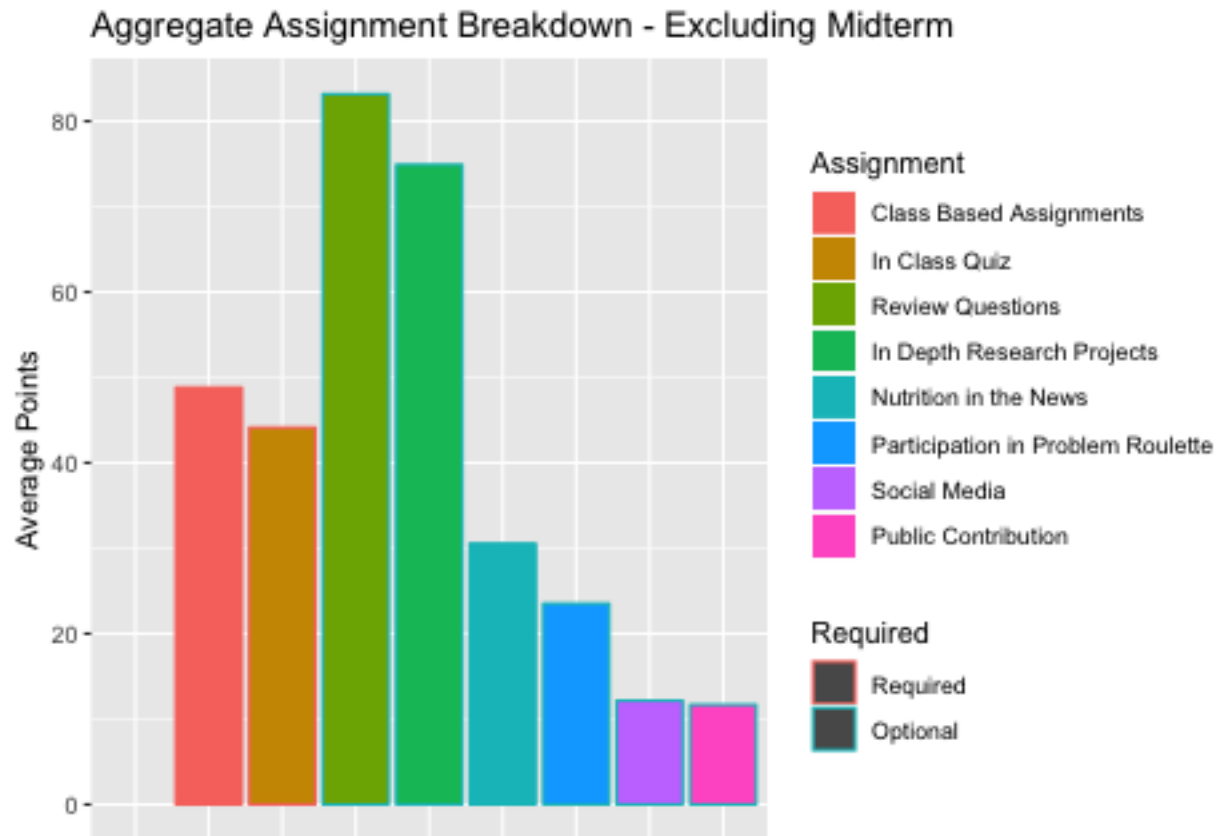
assignment.summary.data$Assignment <- fct_relevel(assignment.summary.data$Assignment, positions)
assignment.summary.data.relative$Assignment <- fct_relevel(assignment.summary.data.relative$Assignment,

p <- ggplot(assignment.summary.data, aes(y=Mean.Points, x=Assignment))
p + geom_bar(stat='identity', aes(fill=Assignment,col=Required)) + scale_x_discrete(limits = positions)
  labs(title="Aggregate Assignment Breakdown", y = "Average Points") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())

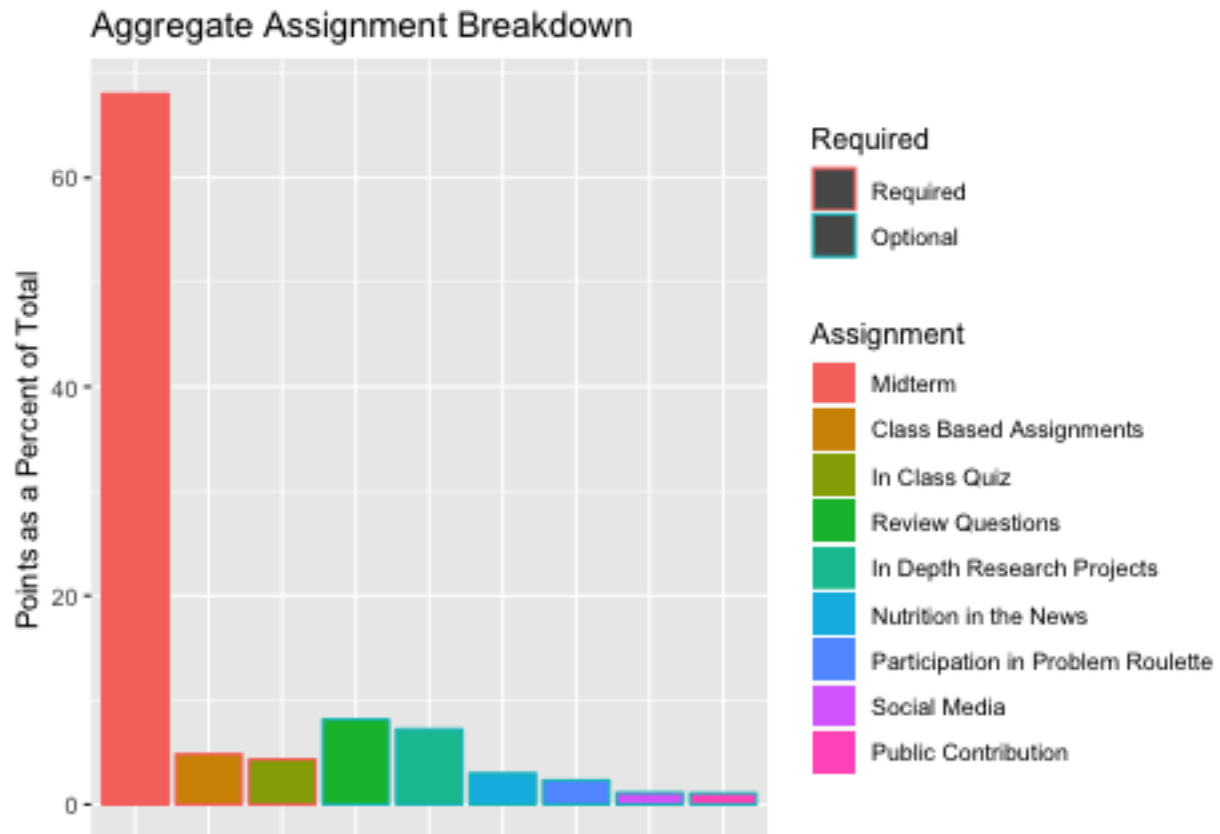
```



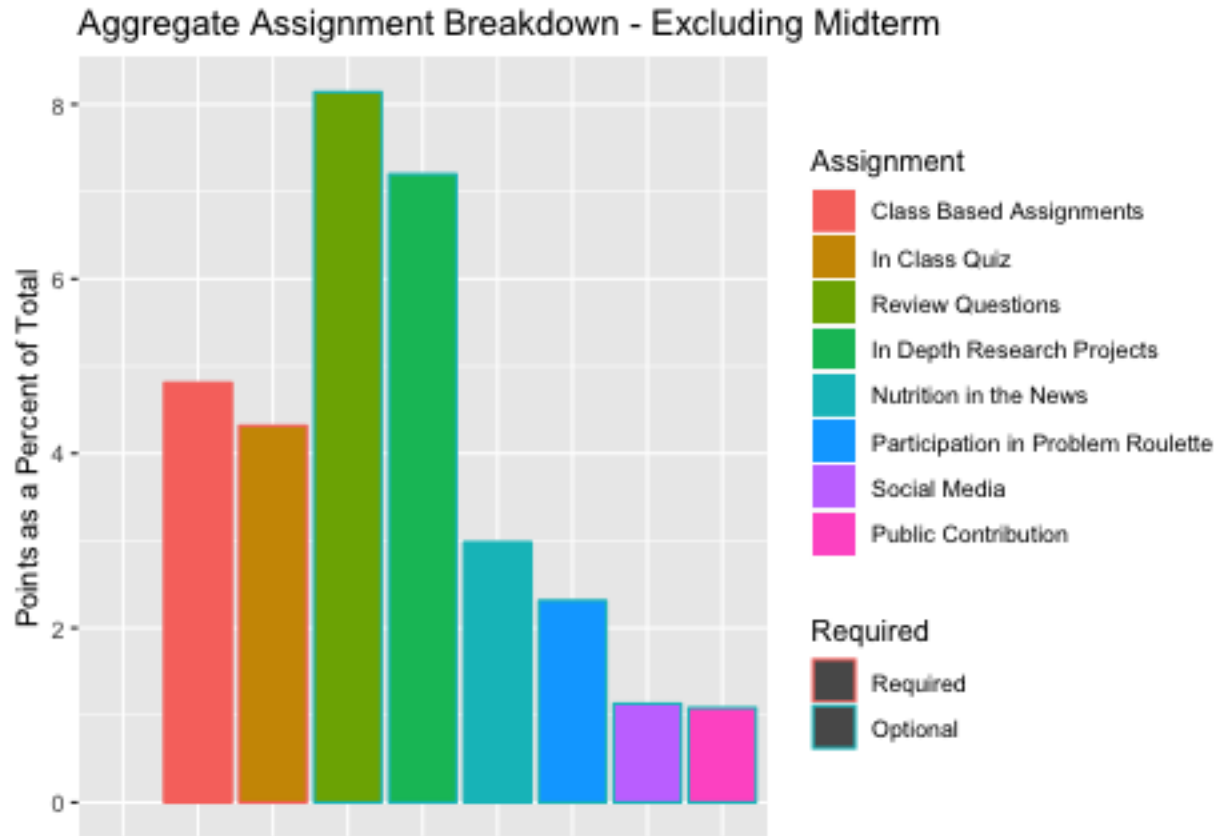
```
p <- ggplot(filter(assignment.summary.data, Assignment != 'Midterm'), aes(y=Mean.Points, x=Assignment))
p + geom_bar(stat='identity', aes(fill=Assignment,col=Required)) + scale_x_discrete(limits = positions)
labs(title="Aggregate Assignment Breakdown - Excluding Midterm", y = "Average Points") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```



```
p <- ggplot(assignment.summary.data.relative, aes(y=Relative.Points, x=Assignment))
p + geom_bar(stat='identity', aes(fill=Assignment,col=Required)) + scale_x_discrete(limits = positions)
labs(title="Aggregate Assignment Breakdown", y = "Points as a Percent of Total") +
theme(axis.title.x=element_blank(),
      axis.text.x=element_blank(),
      axis.ticks.x=element_blank())
```



```
p <- ggplot(filter(assignment.summary.data.relative, Assignment != 'Midterm'), aes(y=Relative.Points, x=Assignment))
p + geom_bar(stat='identity', aes(fill=Assignment,col=Required)) + scale_x_discrete(limits = positions)
labs(title="Aggregate Assignment Breakdown - Excluding Midterm", y = "Points as a Percent of Total")
theme(axis.title.x=element_blank(),
      axis.text.x=element_blank(),
      axis.ticks.x=element_blank())
```



```
kable(assignment.summary.data.required, caption="Average total points")
```

Table 1: Average total points

Required	Median.Points	Median.Points.SD	Relative.Points	Relative.Points.SD
Required	799	94.7	77.8	9.22
Optional	228	69.1	22.2	6.73

```
kable(assignment.summary.data, caption="Average total points per assignment")
```

Table 2: Average total points per assignment

Assignment	Mean.Points	SD.Points	Required
Midterm	699.3	91.72	Required
Class Based Assignments	48.8	2.10	Required
In Class Quiz	44.1	4.56	Required
Review Questions	83.2	22.84	Optional
In Depth Research Projects	74.9	47.25	Optional
Nutrition in the News	30.5	20.60	Optional
Participation in Problem Roulette	23.5	9.34	Optional
Social Media	12.2	17.78	Optional
Public Contribution	11.7	23.19	Optional

```
kable(assignment.summary.data.relative, caption="Average Relative points per assignment as a percent of
```

Table 3: Average Relative points per assignment as a percent of total

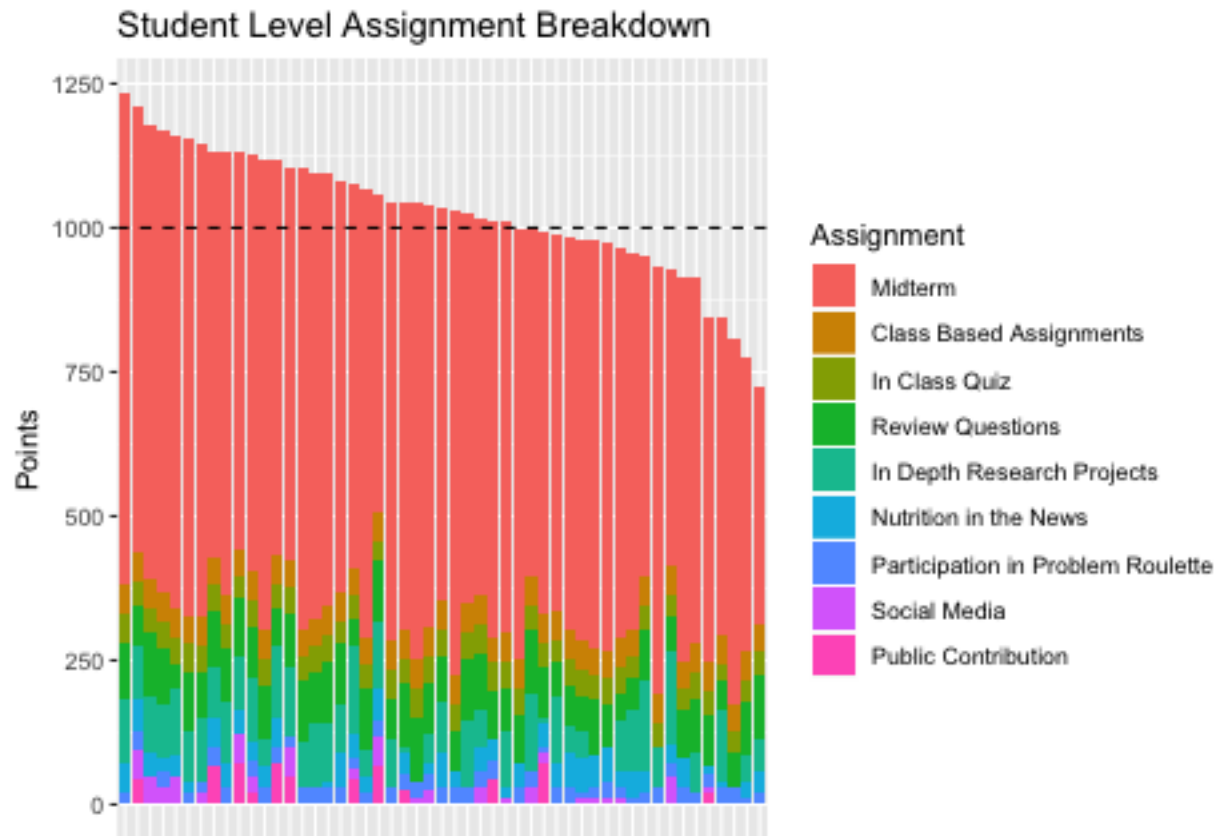
Assignment	Relative.Points	SD.Relative.Points	Required
Midterm	68.02	5.766	Required
Class Based Assignments	4.81	0.633	Required
In Class Quiz	4.32	0.450	Required
Review Questions	8.14	2.371	Optional
In Depth Research Projects	7.20	4.604	Optional
Nutrition in the News	2.98	2.020	Optional
Participation in Problem Roulette	2.31	0.921	Optional
Social Media	1.13	1.629	Optional
Public Contribution	1.08	2.142	Optional

Student Level

```
assessment.dataset$Username <- fct_relevel(assessment.dataset$Username, summary.dataset$Username)
assessment.dataset$Assignment <- fct_relevel(assessment.dataset$Assignment, positions)

p <- ggplot(assessment.dataset, aes(y=Points,x=Username))

p + geom_bar(stat='identity', aes(fill=Assignment)) +
  labs(title="Student Level Assignment Breakdown") +
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) +
  geom_hline(aes(yintercept=1000), color="black", linetype="dashed", size=0.5)
```

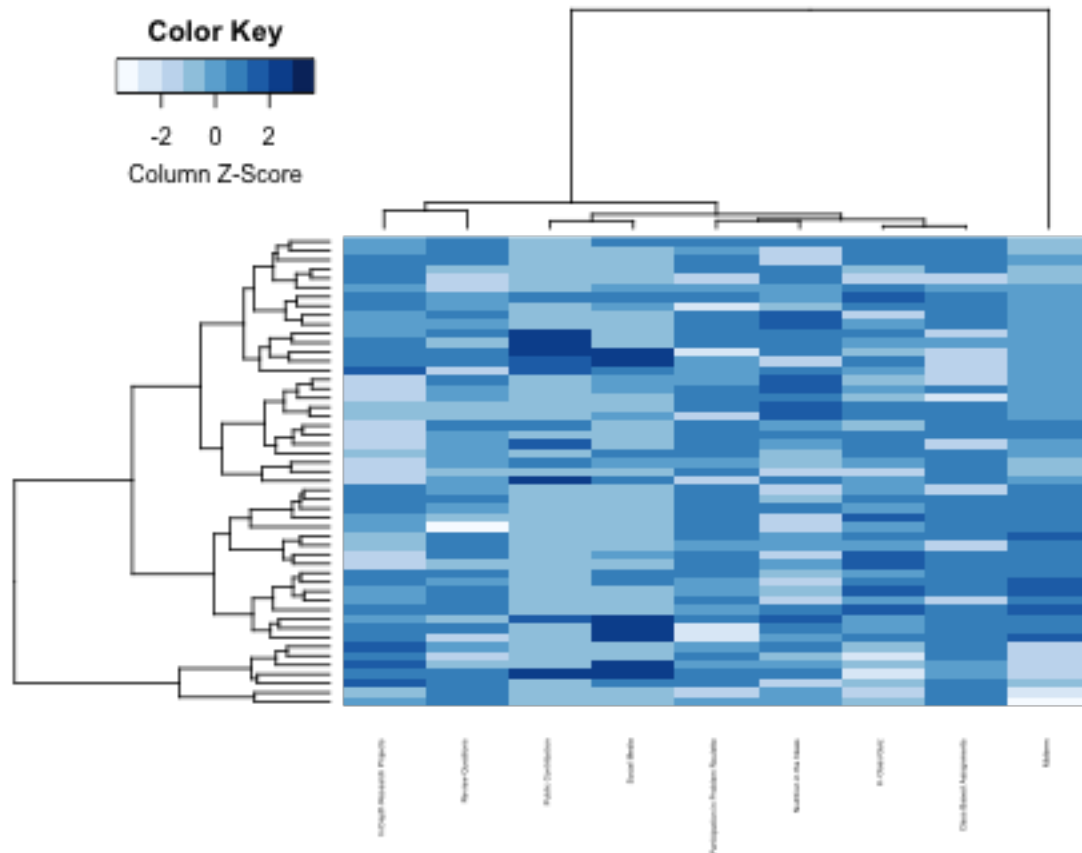



Student Level Paths

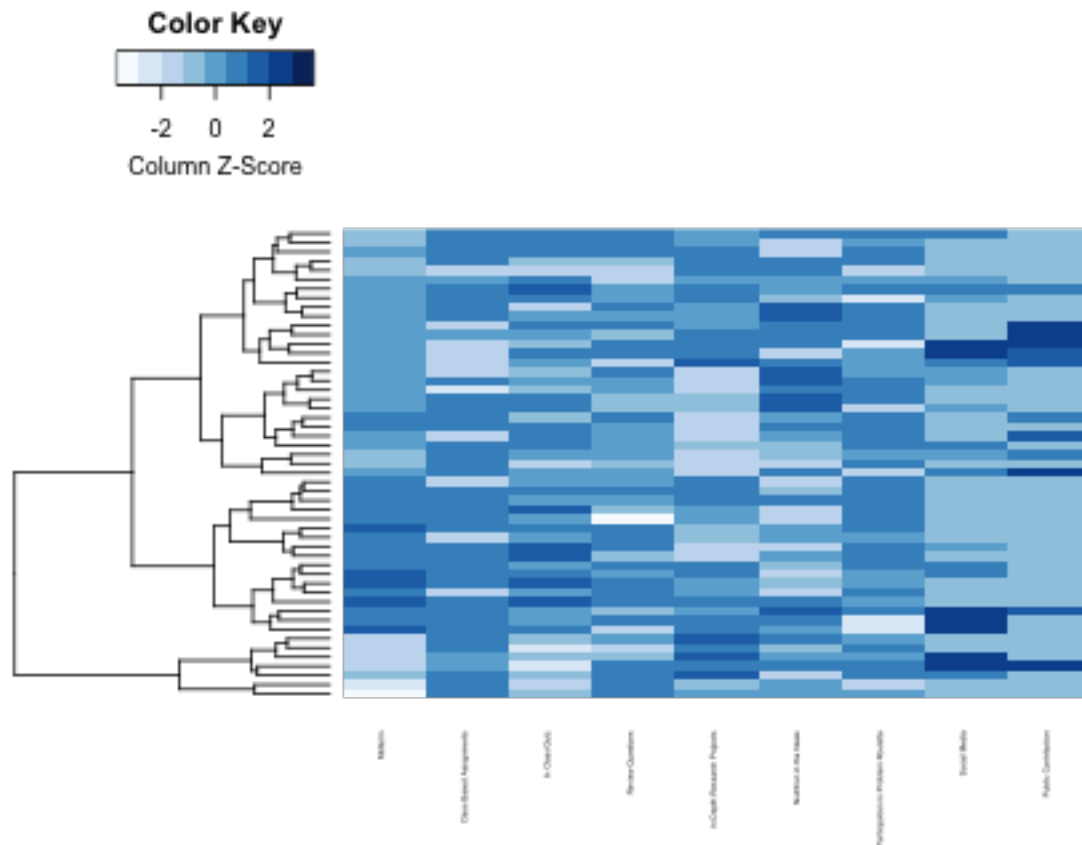
```
assignment.matrix <- assessment.dataset %>% spread(Assignment,Points)
assignment.matrix.names <- assignment.matrix$Username

assignment.matrix <-
  assignment.matrix %>%
  select(-Username) %>%
  as.matrix()

library(RColorBrewer)
library(gplots)
heatmap.2(assignment.matrix,
  scale='col',
  labRow=NA,
  cexCol = 0.3,
  col=brewer.pal(9,"Blues"),
  density.info='none', trace='none')
```



```
heatmap.2(assignment.matrix,
  scale='col',
  labRow=NA,
  Colv=NA,
  cexCol = 0.3,
  col=brewer.pal(9,"Blues"),
  density.info='none', trace='none')
```

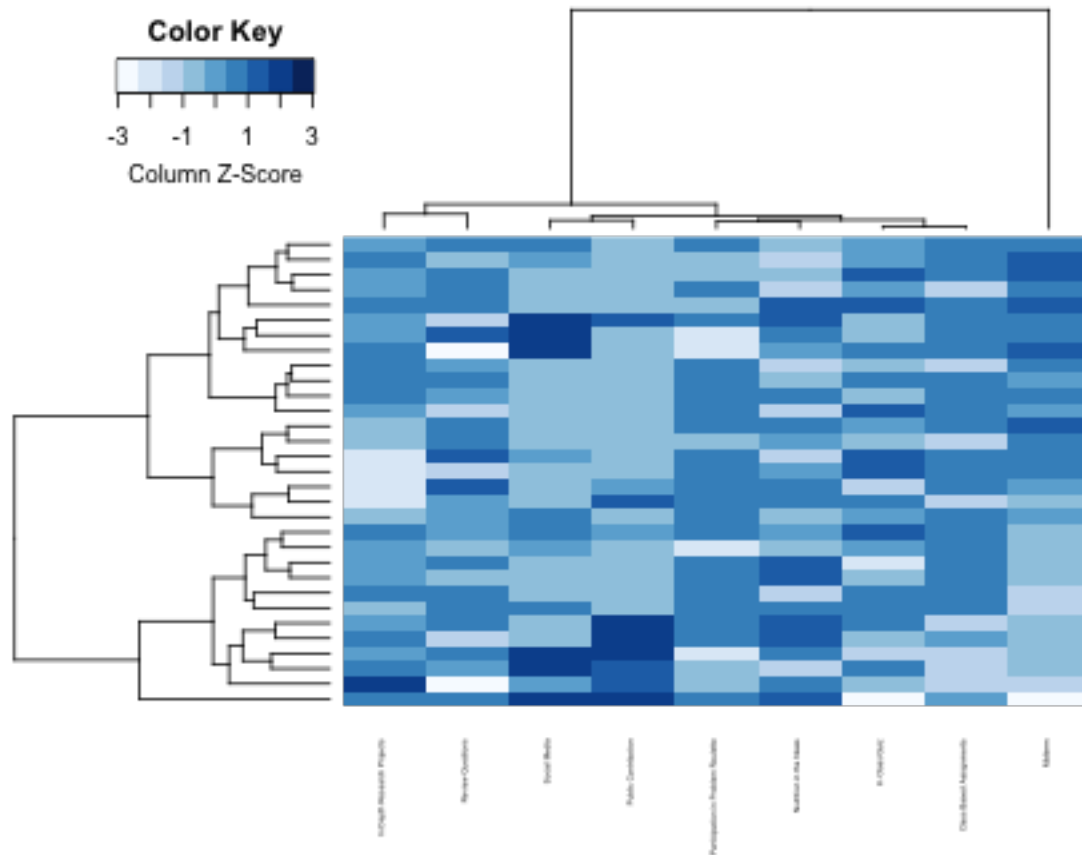


Among Students with an A

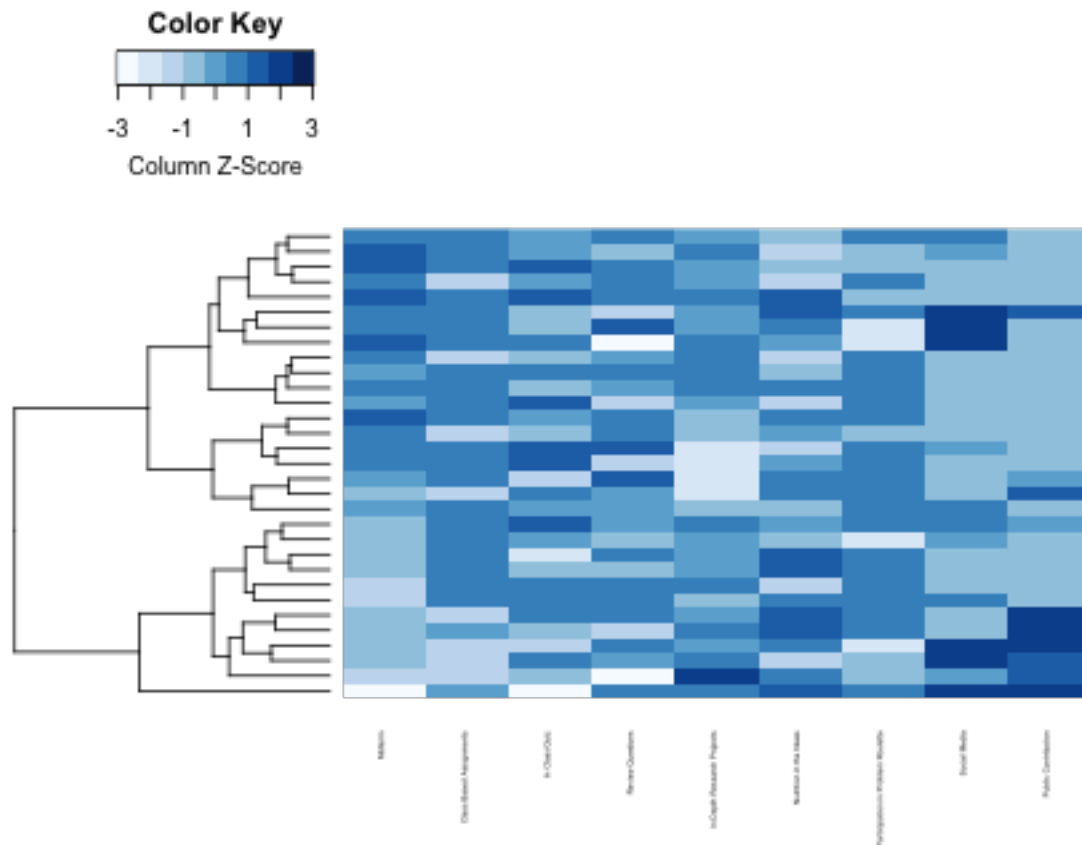
```
assignment.matrix.a <- assessment.dataset %>% spread(Assignment,Points)
assignment.matrix.a$Total <- rowSums(assignment.matrix.a %>% select(-Username))

assignment.matrix.a <-
  assignment.matrix.a %>%
  filter(Total>1000) %>%
  select(-Username, -Total) %>%
  as.matrix()

heatmap.2(assignment.matrix.a,
  scale='col',
  labRow=NA,
  cexCol = 0.3,
  col=brewer.pal(9,"Blues"),
  density.info='none', trace='none')
```

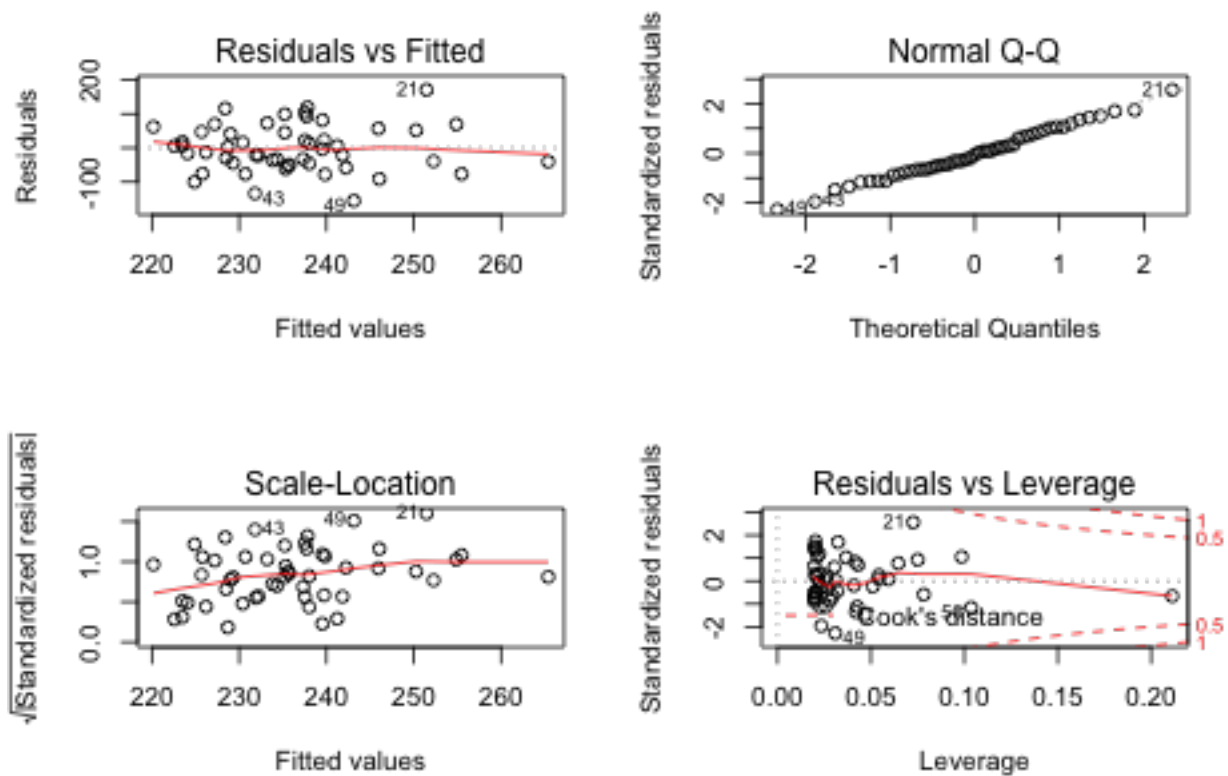


```
heatmap.2(assignment.matrix.a,
  scale='col',
  labRow=NA,
  Colv=NA,
  cexCol = 0.3,
  col=brewer.pal(9,"Blues"),
  density.info='none', trace='none')
```



```
assessment.dataset.required <-
  assessment.dataset %>%
  mutate(Required = ifelse(Assignment %in% required.assignments, "Required", "Optional")) %>%
  group_by(Username, Required) %>%
  summarise(Points = sum(Points)) %>%
  spread(Required, Points) %>%
  mutate(Total.Points = Required + Optional) %>%
  mutate(A.Grade = Total.Points>1000)

lm.points <- lm(Optional ~ Required, data=assessment.dataset.required)
par(mfrow=c(2,2))
plot(lm.points)
```



```
library(broom)
kable(tidy(lm.points), caption="Linear model of optional points dependent on required points per student")
```

Table 4: Linear model of optional points dependent on required points per student.

term	estimate	std.error	statistic	p.value
(Intercept)	315.4	82.356	3.830	0.000
Required	-0.1	0.103	-0.971	0.336

```
shapiro.tidy <- function(x) tidy(shapiro.test(x))

kable(with(assessment.dataset.required, rbind(Optional = shapiro.tidy(Optional),
                                              Required=shapiro.tidy(Required),
                                              Total.Points = shapiro.tidy(Total.Points))), caption="Shapiro-Wilk tests for normality")
```

Table 5: Shapiro-Wilk tests for normality

	statistic	p.value	method
Optional	0.986	0.785	Shapiro-Wilk normality test
Required	0.953	0.042	Shapiro-Wilk normality test
Total.Points	0.970	0.225	Shapiro-Wilk normality test

```
#correlations
```

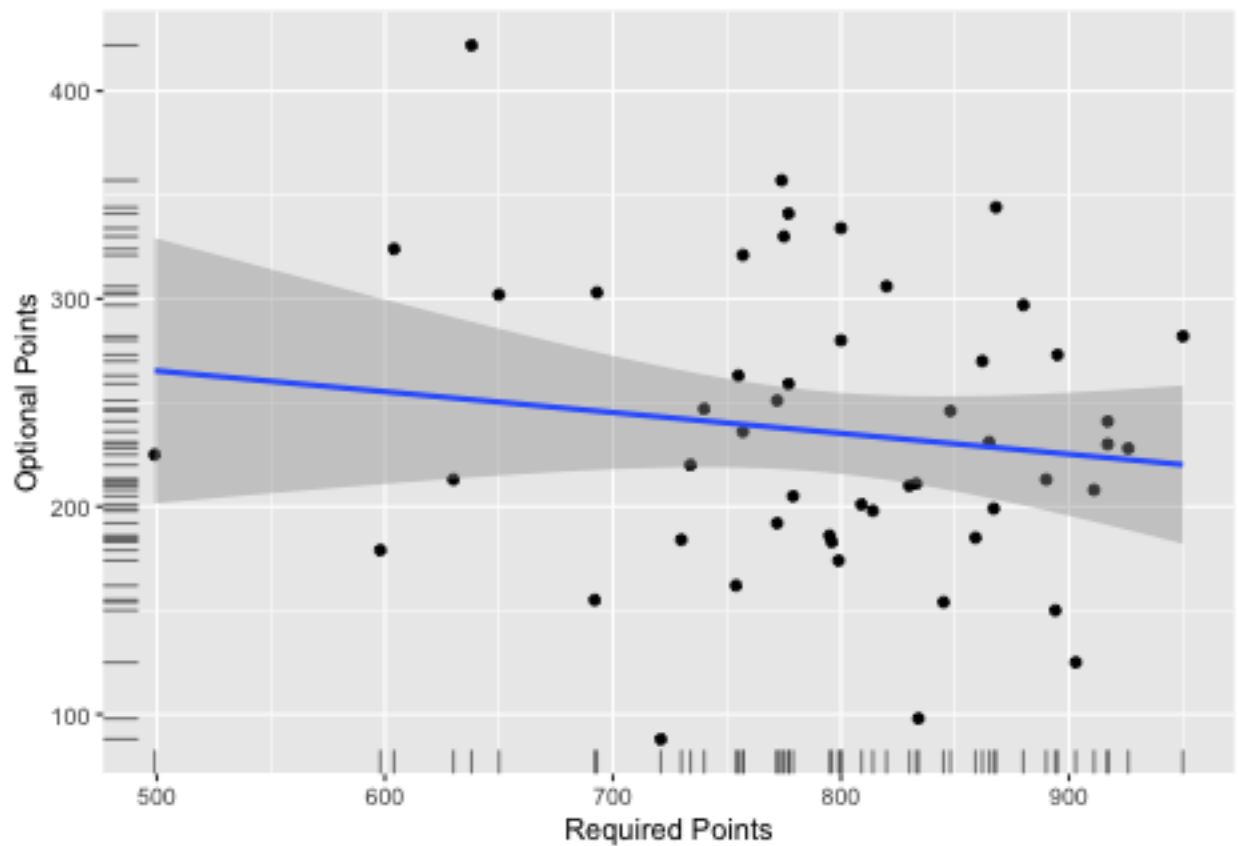
```
kable(tidy(with(assessment.dataset.required, cor.test(Optional, Required,method="spearman"))), caption="")
```

estimate	statistic	p.value	method	alternative
-0.07	23657	0.623	Spearman's rank correlation rho	two.sided

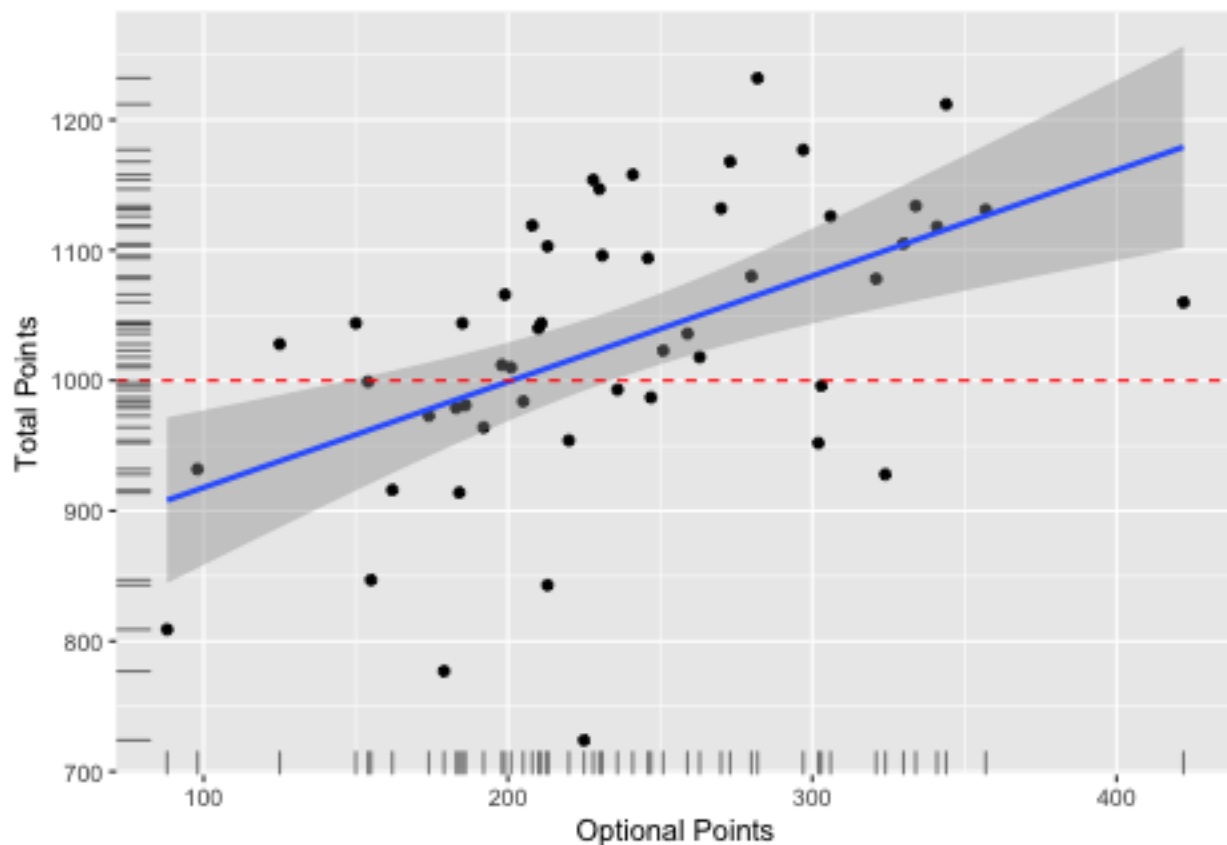
```
kable(tidy(with(assessment.dataset.required, cor.test(Optional, Total.Points,method="pearson"))), caption="")
```

estimate	statistic	p.value	parameter	conf.low	conf.high	method	alternative
0.513	4.18	0	49	0.277	0.691	Pearson's product-moment correlation	two.sided

```
p <- ggplot(assessment.dataset.required, aes(y=Optional,x=Required))
p + geom_point() +
  geom_smooth(method='lm') +
  labs(y="Optional Points", x="Required Points") +
  geom_rug(alpha=0.5)
```



```
p <- ggplot(assessment.dataset.required, aes(y=Total.Points,x=Optional))
p + geom_point() +
  geom_smooth(method='lm') +
  labs(y="Total Points", x="Optional Points") +
  geom_hline(yintercept = 1000, lty=2,col="red") +
  geom_rug(alpha=0.5)
```



Likelihood of an A Grade

```
glm.a.grade <- glm(A.Grade ~ Optional, data=assessment.dataset.required, family="binomial")
tidy(glm.a.grade) %>% kable(caption="Logistic regression of optional assignments vs grade")
```

Table 8: Logistic regression of optional assignments vs grade

term	estimate	std.error	statistic	p.value
(Intercept)	-2.995	1.299	-2.31	0.021
Optional	0.015	0.006	2.65	0.008

```
#coef as logit

predict.data <- data.frame(Optional=seq(0,400),by=5)
predict.data$Logit <- predict(glm.a.grade, newdata=predict.data)
predict.data <-
  predict.data %>%
  mutate(Odds = exp(Logit)) %>%
  mutate(Probability = Odds/(1+Odds))

ggplot(predict.data, aes(y=Probability,x=Optional)) +
```

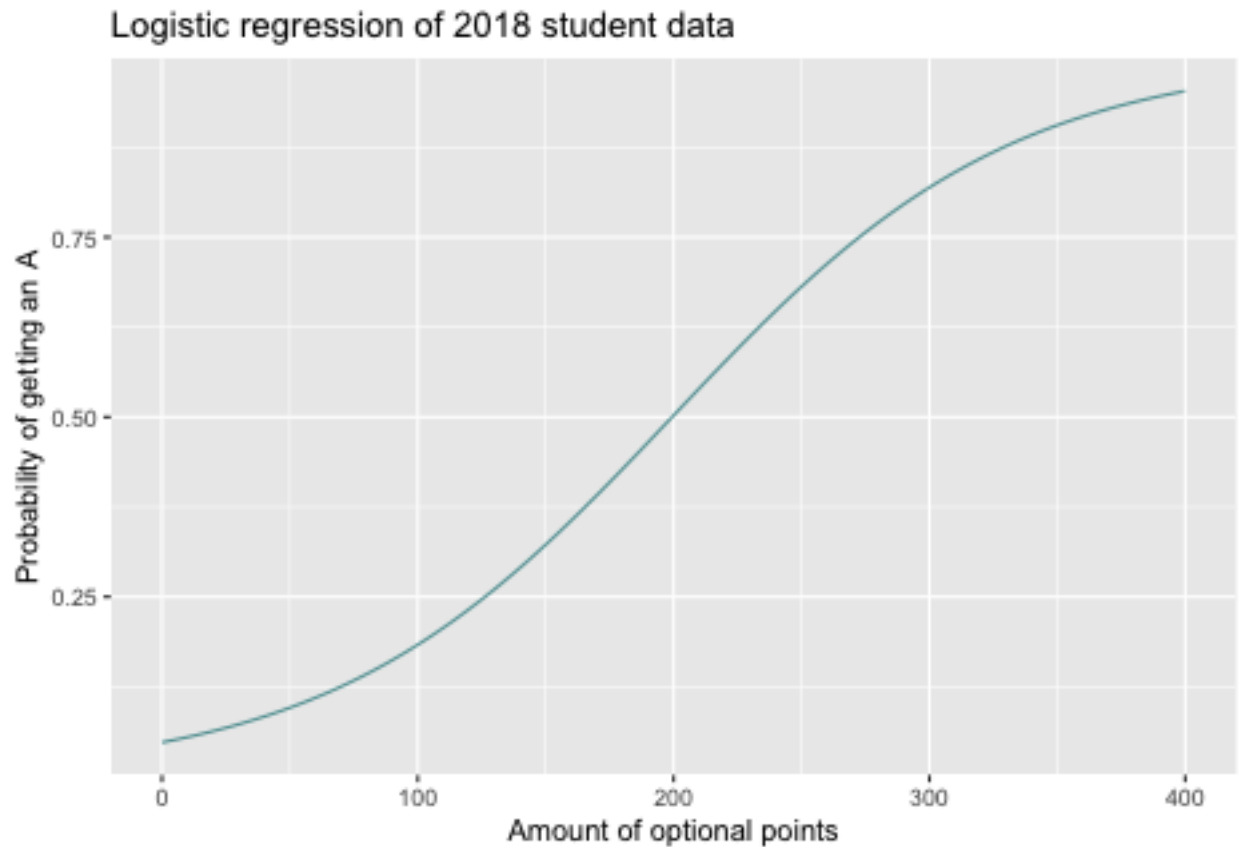



Figure 1: Logistic regression for points required to get an A

```
geom_line(color='cadetblue') +
labs(y="Probability of getting an A",
     x="Amount of optional points",
     title="Logistic regression of 2018 student data")
```

```
beta.a.grade <- coef(glm.a.grade)['Optional'] #coef as logit
odds.a.grade <- exp(beta.a.grade) #coef as odds
prob.a.grade <- odds.a.grade/(1 + odds.a.grade)
```

Based on last year,

Session Information

```
sessionInfo()
```

```
## R version 3.5.0 (2018-04-23)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
```

```

## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] broom_0.5.0      gplots_3.0.1      RColorBrewer_1.1-2
## [4] forcats_0.3.0    ggplot2_3.0.0      bindrcpp_0.2.2
## [7] tidyr_0.8.1      dplyr_0.7.6        readr_1.1.1
## [10] knitr_1.20
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.18      highr_0.7          pillar_1.3.0
## [4] compiler_3.5.0    plyr_1.8.4         bindr_0.1.1
## [7] bitops_1.0-6      tools_3.5.0        digest_0.6.16
## [10] lattice_0.20-35   nlme_3.1-137       evaluate_0.11
## [13] tibble_1.4.2      gtable_0.2.0       pkgconfig_2.0.2
## [16] rlang_0.2.2       cli_1.0.0          yaml_2.2.0
## [19] withr_2.1.2       stringr_1.3.1      caTools_1.17.1.1
## [22] gtools_3.8.1      hms_0.4.2          rprojroot_1.3-2
## [25] grid_3.5.0        tidyselect_0.2.4   glue_1.3.0
## [28] R6_2.2.2          fansi_0.3.0        rmarkdown_1.10
## [31] gdata_2.18.0      purrr_0.2.5        magrittr_1.5
## [34] backports_1.1.2    scales_1.0.0       htmltools_0.3.6
## [37] assertthat_0.2.0  colorspace_1.3-2   labeling_0.3
## [40] KernSmooth_2.23-15 utf8_1.1.4         stringi_1.2.4
## [43] lazyeval_0.2.1    munsell_0.5.0      crayon_1.3.4

```