141XP_FinalProjectKatie

Katie Munteanu

2024-05-08

```
suppressWarnings(library(tidyverse))
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

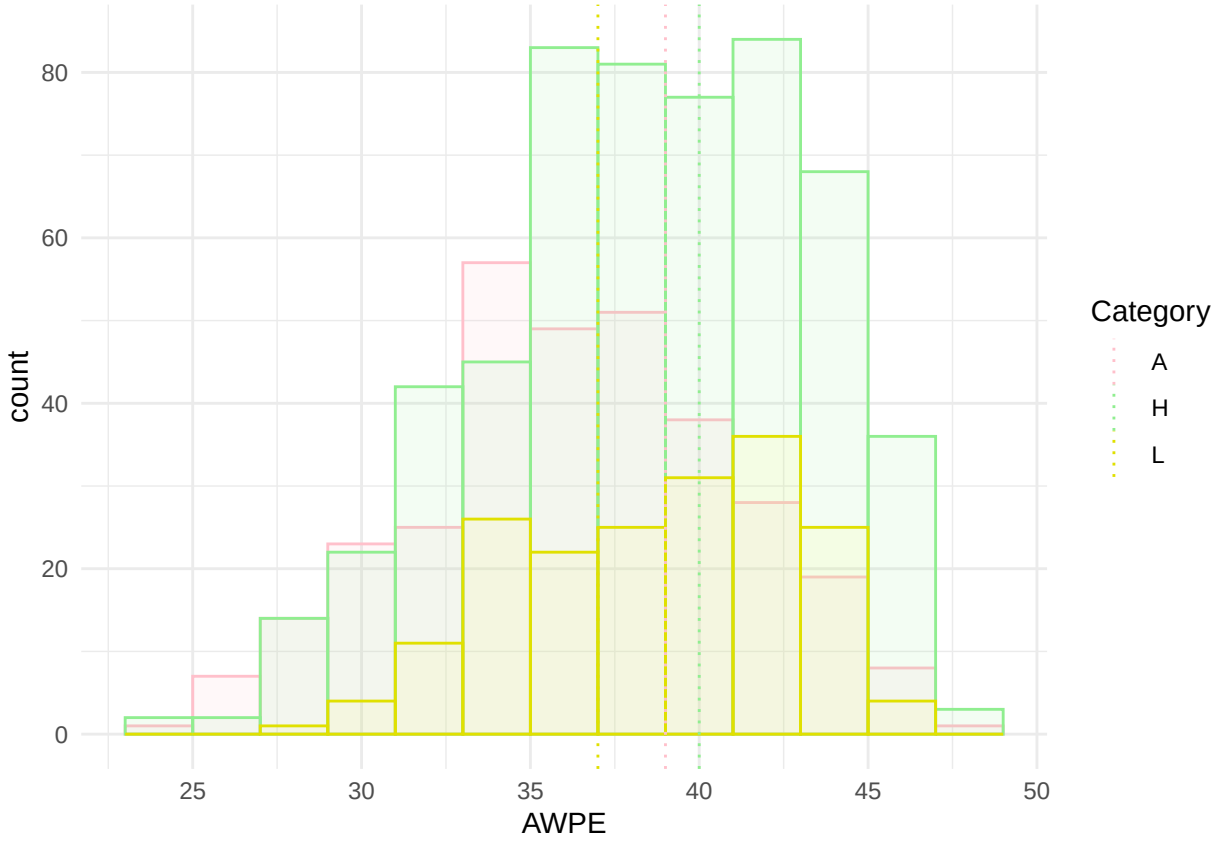
```
english_data <- read.csv("English_Data_Clean.csv")
```

Messing around with histograms

```
# Calculate median for each category
median_data <- english_data %>%
  group_by(Category) %>%
  summarize(median_AWPE = median(AWPE, na.rm = TRUE))

x_breaks <- seq(5, 50, by = 5)

ggplot(data = english_data) +
  geom_histogram(data = english_data %>% filter(Category == "L"), aes(x = AWPE), fill = "pink", color =
  geom_histogram(data = english_data %>% filter(Category == "A"), aes(x = AWPE), fill = "lightgreen", co
  geom_histogram(data = english_data %>% filter(Category == "H"), aes(x = AWPE), fill = "#FFFF66", colo
  geom_vline(data = median_data, aes(xintercept = median_AWPE, color = Category), linetype = "dotted") +
  scale_color_manual(values = c("#FFC0CB", "#90EE90", "#E0E000")) +
  labs(fill = "Category") +
  scale_x_continuous(breaks = x_breaks) +
  theme_minimal()
```



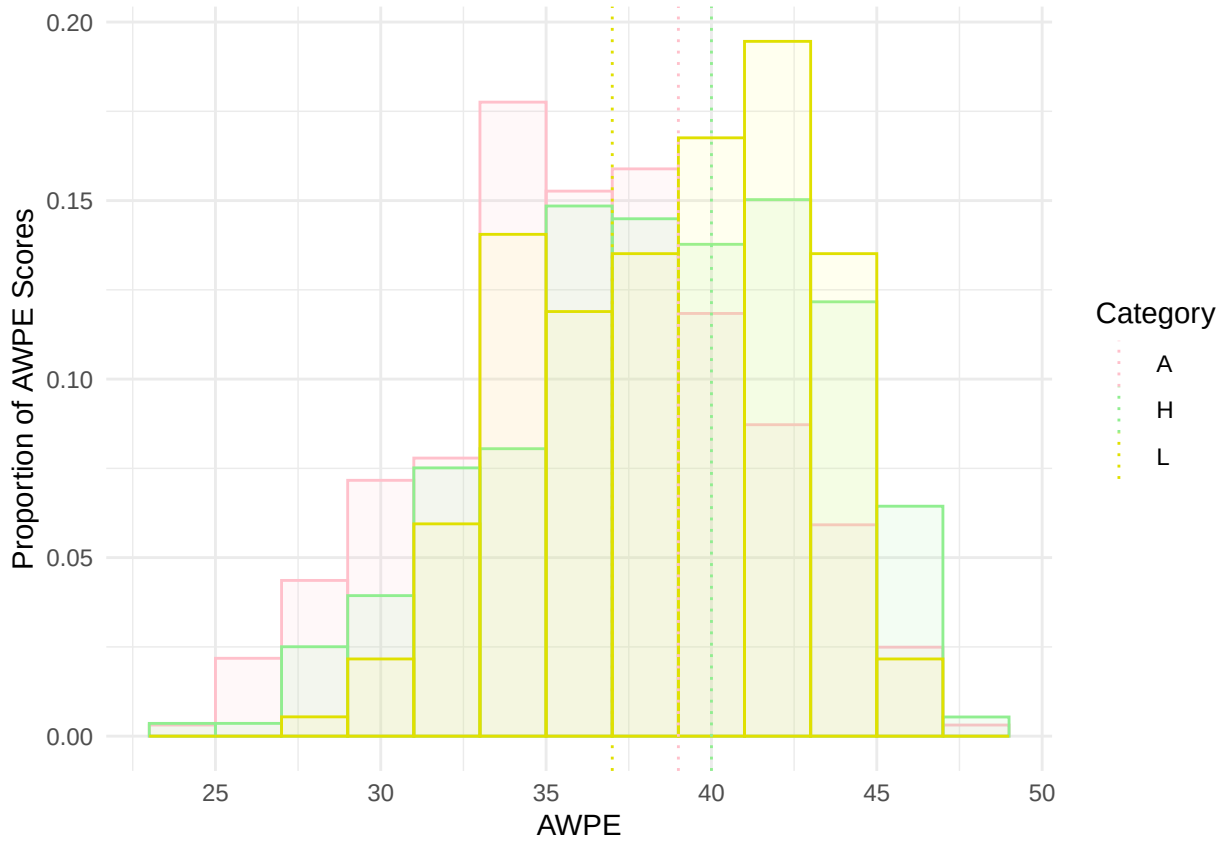
Sorry I didn't include the legend for now, since formatting it can be a lil annoying, but pink = "L", green = "A", yellow = "H" (and for some reason color of the median lines isn't matching up T.T)

```
median_data <- english_data %>%
  group_by(Category) %>%
  summarize(median_AWPE = median(AWPE, na.rm = TRUE))

x_breaks <- seq(5, 50, by = 5)

ggplot(english_data, aes(x = AWPE, y = ..count../sum(..count..), fill = Category)) +
  geom_histogram(data = english_data %>% filter(Category == "L"), aes(x = AWPE), fill = "pink", color =
  geom_histogram(data = english_data %>% filter(Category == "A"), aes(x = AWPE), fill = "lightgreen", c
  geom_histogram(data = english_data %>% filter(Category == "H"), aes(x = AWPE), fill = "#FFFF66", colo
  geom_vline(data = median_data, aes(xintercept = median_AWPE, color = Category), linetype = "dotted") +
  scale_color_manual(values = c("#FFC0CB", "#90EE90", "#E0E000")) +
  labs(fill = "Category") +
  scale_x_continuous(breaks = x_breaks) +
  theme_minimal() + ylab("Proportion of AWPE Scores")
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Ok josh I tried to re-make this histogram using the proportion instead of count. Not sure if this is what you were hoping for.

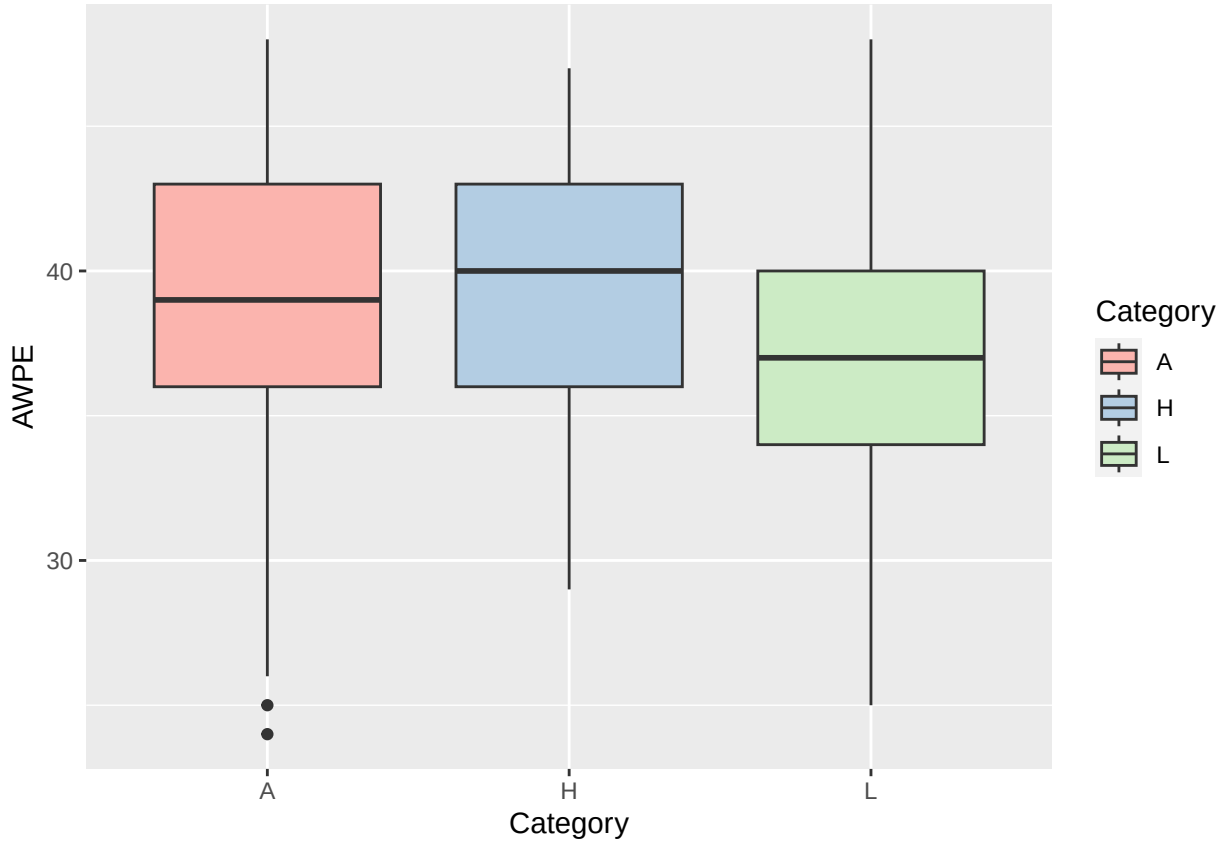
```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.2.3
```

```
## corrplot 0.92 loaded
```

```
library(RColorBrewer)

ggplot(english_data, aes(x = Category, y = AWPE, fill = Category)) +
  geom_boxplot() +
  labs(x = "Category", y = "AWPE") +
  scale_fill_brewer(palette = "Pastel1")
```




```
# Just curious to see if anything is significant.

logistic_model <- glm(Category ~ AWPE, data = english_data_binary)
summary(logistic_model)
```

```
##
## Call:
## glm(formula = Category ~ AWPE, data = english_data_binary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6584  -0.5063   0.3693   0.4660   0.6733
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.005018   0.123615  -0.041    0.968
## AWPE         0.013821   0.003200   4.319 1.71e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2455404)
##
##     Null deviance: 265.59  on 1064  degrees of freedom
## Residual deviance: 261.01  on 1063  degrees of freedom
## AIC: 1530.8
##
```

```
## Number of Fisher Scoring iterations: 2
```

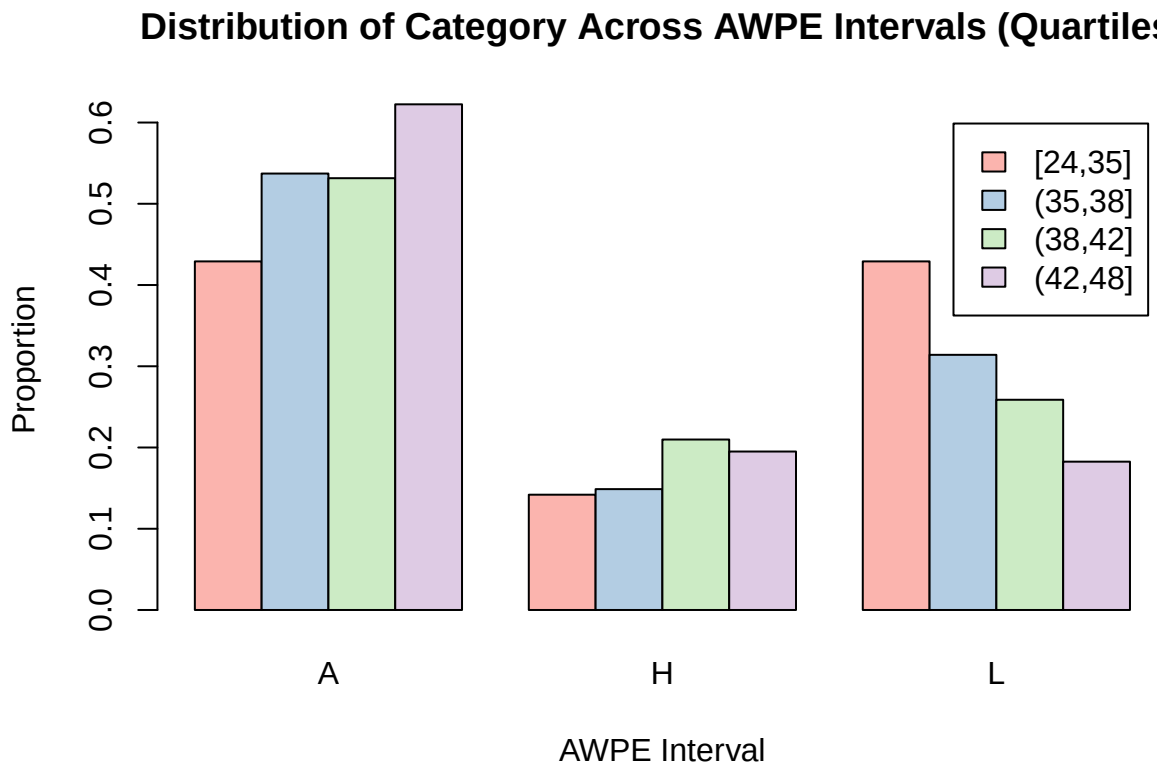
```r
# Define quartiles for AWPE variable
quartile_intervals <- quantile(english_data$AWPE, probs = seq(0, 1, by = 0.25))

# Create intervals based on quartiles
english_data$AWPE_interval <- cut(english_data$AWPE, breaks = quartile_intervals, include.lowest = TRUE)

# Calculate the distribution of Category within each interval
interval_distribution <- table(english_data$AWPE_interval, english_data$Category)

# Convert to proportions for better comparison
interval_distribution_prop <- prop.table(interval_distribution, margin = 1)

# Visualize the relationship using a stacked bar plot
barplot(interval_distribution_prop, beside = TRUE, legend = TRUE, col = brewer.pal(n = 4, name = "Pastel
        main = "Distribution of Category Across AWPE Intervals (Quartiles)",
        xlab = "AWPE Interval", ylab = "Proportion")
```



I'm gonna explore the correlations between quantile and accuracy

```r
library(corrplot)
# Calculate the correlation between A and AWPE_interval for each interval
correlation_intervals <- sapply(levels(english_data$AWPE_interval), function(interval) {
  subset_data <- subset(english_data, AWPE_interval == interval)
  cor_test <- cor.test(subset_data$AWPE, as.numeric(subset_data$Category == "A"), method = "pearson")
```

```
  cor_test$estimate
})

# Print the correlation for each interval
print(correlation_intervals)
```

```
## [24,35].cor (35,38].cor (38,42].cor (42,48].cor
## -0.06227000  0.02222429  0.08715088  0.15905540
```

```
# Nothing seems to be highly correlated, but I'll run a logistic regression and see.
logit_model <- glm(Category == "A" ~ AWPE_interval, data = english_data, family = binomial)
summary(logit_model)
```

```
##
## Call:
## glm(formula = Category == "A" ~ AWPE_interval, family = binomial,
##     data = english_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3957  -1.2314   0.9738   1.1244   1.3009
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.2857     0.1174  -2.433   0.0150 *
## AWPE_interval(35,38]   0.4347     0.1744   2.493   0.0127 *
## AWPE_interval(38,42]   0.4118     0.1668   2.468   0.0136 *
## AWPE_interval(42,48]   0.7855     0.1773   4.429 9.45e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1473.8  on 1064  degrees of freedom
## Residual deviance: 1453.4  on 1061  degrees of freedom
## AIC: 1461.4
##
## Number of Fisher Scoring iterations: 4
```

Ok so [42,48] is the best range? How many observations does that even leave us with?

```
nrow(english_data %>% filter(AWPE >= 42))
```

```
## [1] 312
```

```
312 / nrow(english_data)
```

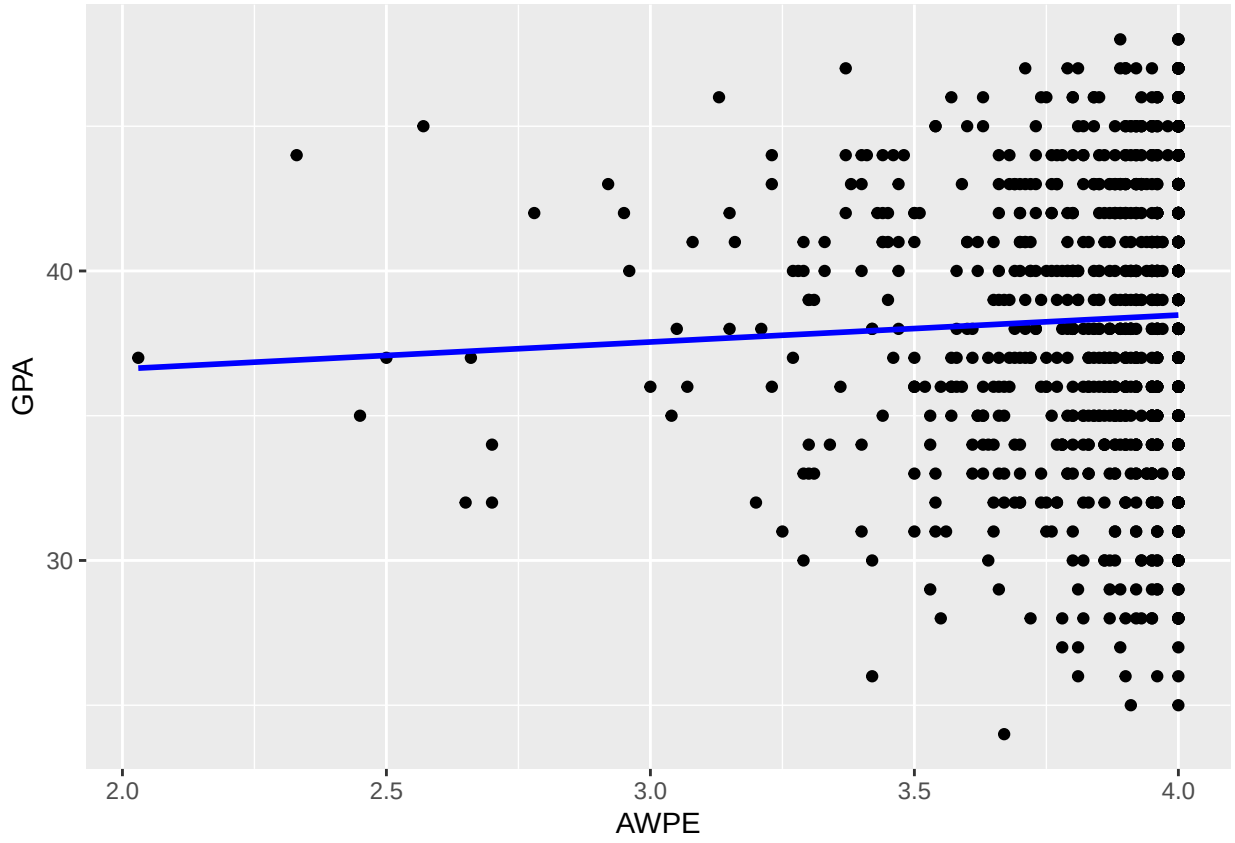
```
## [1] 0.2929577
```

Ok actually not bad. So assuming [42,48] is even reliable, we could potentially shave off 30% of the future essays the raters will have to read through. Assuming our sample is representative of future test-takers...

Ok let's explore GPA a bit.

```
ggplot(english_data, aes(x = GPA, y = AWPE)) +
  geom_point(color = "black") +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +  # Add a linear trend line without confidenc
  labs(x = "AWPE", y = "GPA")  # Add axis labels if necessary
```

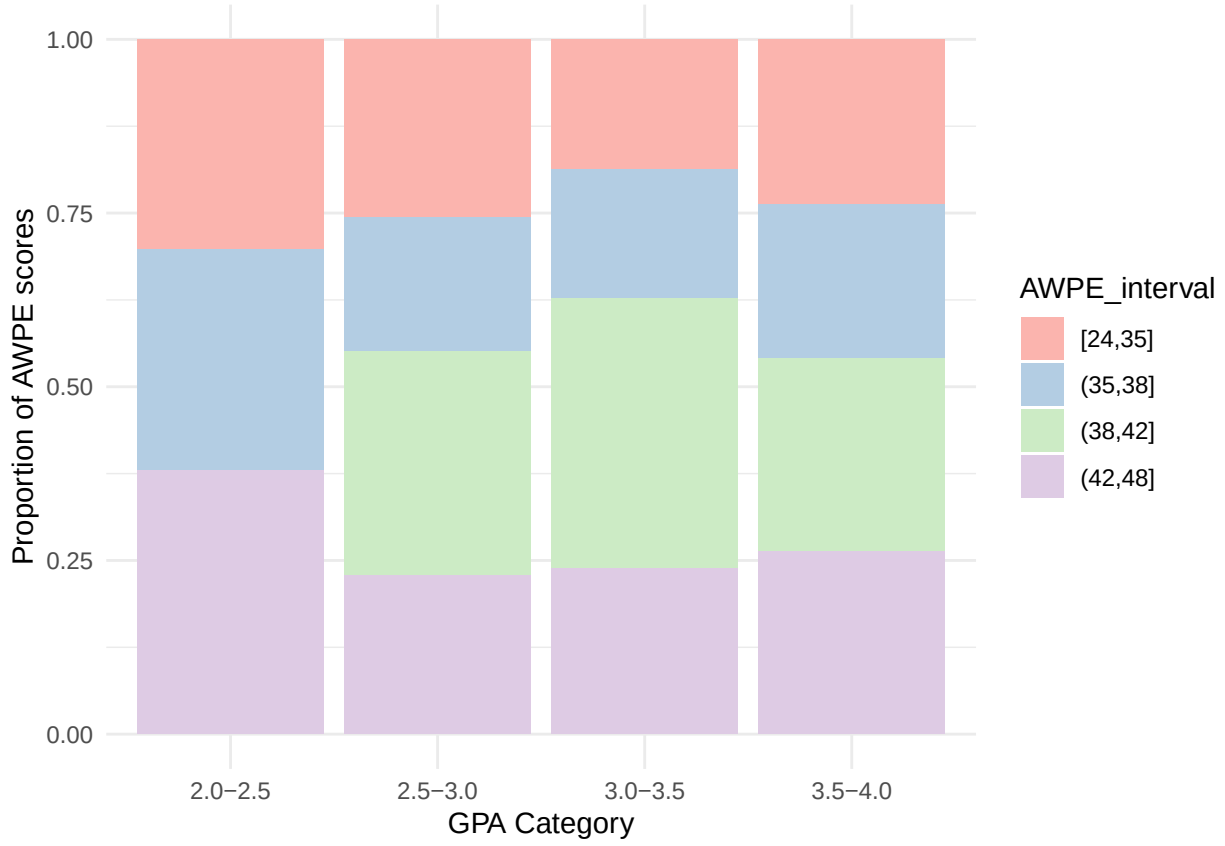
```
## `geom_smooth()` using formula = 'y ~ x'
```



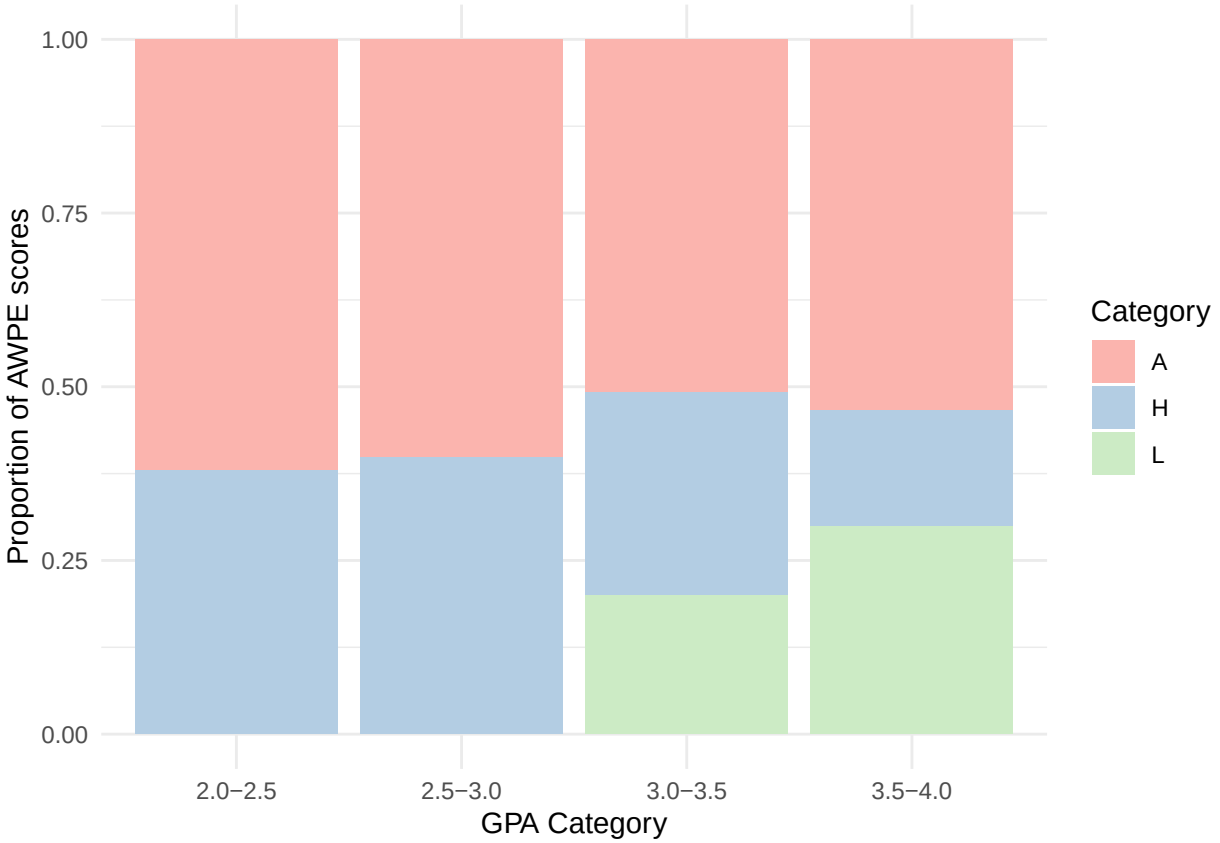
Some stacked barcharts.

```
english_data <- english_data %>%
  mutate(GPA_category = case_when(
    GPA >= 2 & GPA < 2.5 ~ "2.0-2.5",
    GPA >= 2.5 & GPA < 3 ~ "2.5-3.0",
    GPA >= 3 & GPA < 3.5 ~ "3.0-3.5",
    GPA >= 3.5 & GPA <= 4 ~ "3.5-4.0",
  ))

ggplot(english_data, aes(fill = AWPE_interval, x = GPA_category, y = AWPE)) +
  geom_bar(position="fill", stat="identity") +
  labs(x = "GPA Category", y = "Proportion of AWPE scores") +
  theme_minimal() +
  scale_fill_brewer(palette = "Pastel1")
```




```
ggplot(english_data, aes(fill=Category, y=AWPE, x=GPA_category)) +
    geom_bar(position="fill", stat="identity") +
  scale_fill_brewer(palette = "Pastel1") +
  theme_minimal() +
  labs(x = "GPA Category", y = "Proportion of AWPE scores")
```



So what I'm inferring from these graphs is that there's definitley a relationship between AWPE score and GPA. The lower your GPA is, the more confident you are, hence the higher proportion of scores in the 42,48 interval. Then in the second graph you can see that the there were the most accurate self-placements in students with 2.0-2.5 GPA.

What I don't know is whether it's GPA that's the true predictor the self-placement accuracy, or if it's the AWPE interval.