Teen Gambling in Britain

Tyler Frankenberg

1/30/2022

Import packages

```
library(tidyverse)
```

Import data & basic cleaning/ transformations

The `teengamb` dataset from package `faraway` includes data from a 1988 survey on teenage gambling in Britain. Per *RDocumentation*, it includes the following columns:

- `sex` : 0=male, 1=female
- `status` : socioeconomic status score based on parents' occupation
- `income` : weekly income in GBP
- `verbal` : verbal score of the number of words out of 12 correctly defined
- `gamble` : gambling activity (weighted score combining annual gambling frequency and amount wagered)

The data was collected via a survey conducted in two classrooms of 13-14 year olds in Exeter, England

Sources: https://www.rdocumentation.org/packages/faraway/versions/1.0.7/topics/ (https://www.rdocumentation.org/packages/faraway/versions/1.0.7/topics/) https://www.researchgate.net/publication/226877934_Gambling_in_young_adolescents (https://www.researchgate.net/publication/226877934_Gambling_in_young_adolescents)

Dataset transformations

We're going to do some simple transformations on this dataset.
- first, we'll replace the binary values of sex with their corresponding label; either "male" or "female"
- second, we'll change this column's type to factor
- finally, we'll select only the columns we need for the analysis

```
url <- "https://raw.githubusercontent.com/curdferguson/data621/main/datasets/teengamb.txt"

teengamb <- read_tsv(url, skip = 1, col_names = c("index", "sex1", "status", "income", "verbal",
"gamble"), show_col_types=FALSE)

teengamb <- teengamb %>%
    mutate(sex = case_when(
      teengamb$sex1 == 0 ~ "male",
      teengamb$sex1 == 1 ~ "female"))

teengamb$sex <- teengamb$sex %>% as_factor()

teengamb <- teengamb[,3:7]
```

Glimpse dataset structure and each column’s summary statistics

We’ll glimpse the dataset’s structure by viewing its first five rows, as well as a brief summary of each column’s distribution, and histograms for each of the numerical datasets.

The most important finding from our initial analysis is that the `gamble` variable - which is going to be the target in our linear model - is heavily right-skewed; it seems to follow an exponential distribution.

We’ll correct for this in our linear model by creating an additional column with the base-10 logarithm of `gamble` .

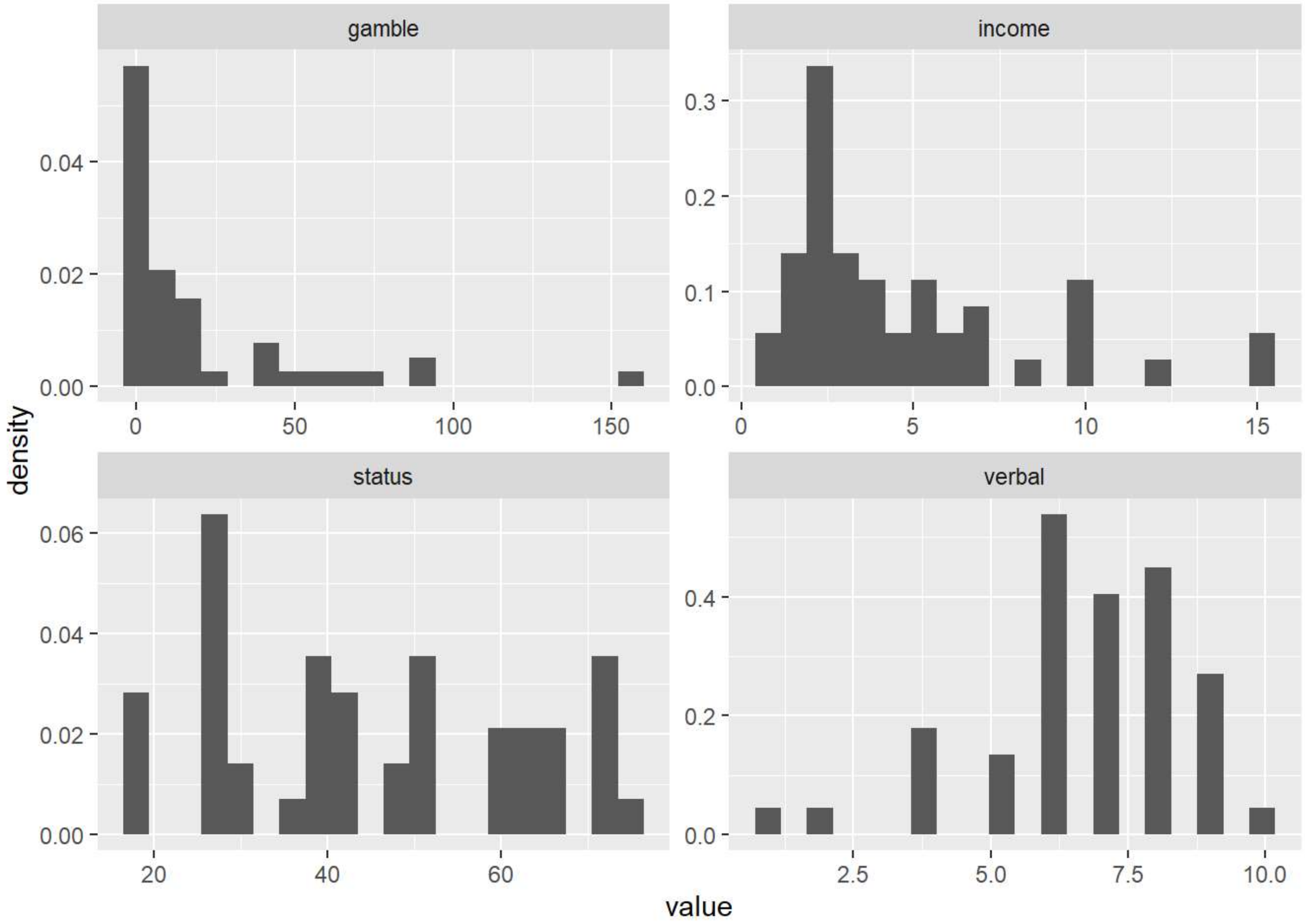
```
teengamb %>% head(5)
```

```
## # A tibble: 5 x 5
##   status income verbal gamble sex
##    <dbl>  <dbl>  <dbl>  <dbl> <fct>
## 1     51    2        8    0   female
## 2     28    2.5      8    0   female
## 3     37    2        6    0   female
## 4     28    7        4    7.3 female
## 5     65    2        8   19.6 female
```

```
teengamb %>% summary()
```

```
##      status          income           verbal          gamble          sex
##  Min.   :18.00   Min.   : 0.600   Min.   : 1.00   Min.   :  0.0   female:19
##  1st Qu.:28.00   1st Qu.: 2.000   1st Qu.: 6.00   1st Qu.:  1.1   male  :28
##  Median :43.00   Median : 3.250   Median : 7.00   Median :  6.0
##  Mean   :45.23   Mean   : 4.642   Mean   : 6.66   Mean   : 19.3
##  3rd Qu.:61.50   3rd Qu.: 6.210   3rd Qu.: 8.00   3rd Qu.: 19.4
##  Max.   :75.00   Max.   :15.000   Max.   :10.00   Max.   :156.0
```

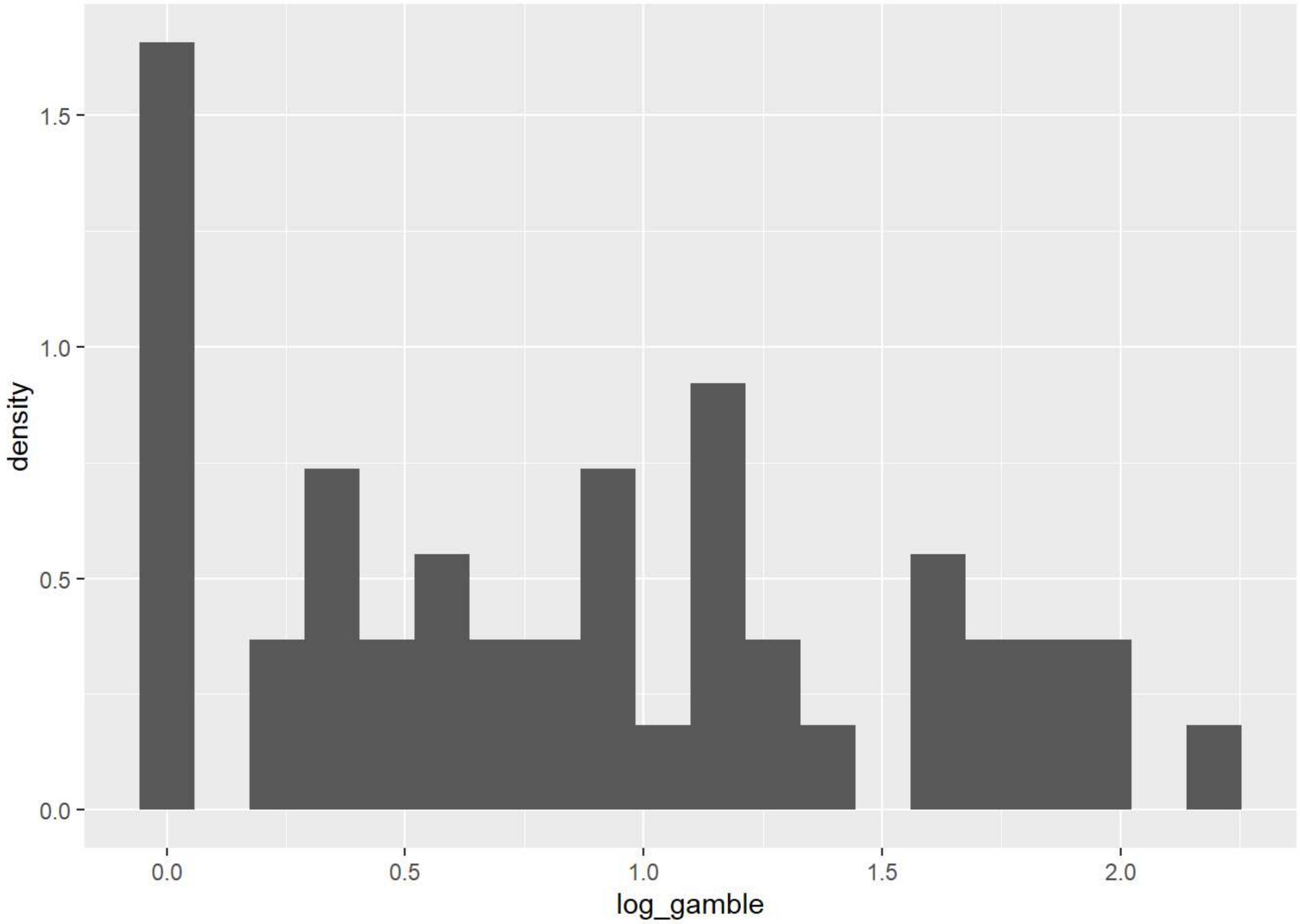
```
cols_list <- colnames(teengamb)

teengamb_numeric <- teengamb %>% select(where(is.numeric))
teengamb_numeric_long <- teengamb_numeric %>% pivot_longer(colnames(teengamb_numeric)) %>% as.da
ta.frame()

ggplot(data=teengamb_numeric_long, aes(x=value)) +
  geom_histogram(aes(y=..density..), bins=20) +
  facet_wrap(~ name, scales = "free")
```



Apply log transformation to `gamble`

Since the base-10 logarithm of 0 results in a value of negative 1, which we won't be able to pass to our `lm` function, we're going to apply an additional transformation to allow for compatibility with the log transformation. We will add 1 to each of the values in `teengamb$gamble` by 10 before taking the logarithm.

```
teengamb <- teengamb %>% mutate(log_gamble = log10((teengamb$gamble + 1)))

ggplot(data=teengamb, aes(x=log_gamble)) +
  geom_histogram(aes(y=..density..), bins=20)
```

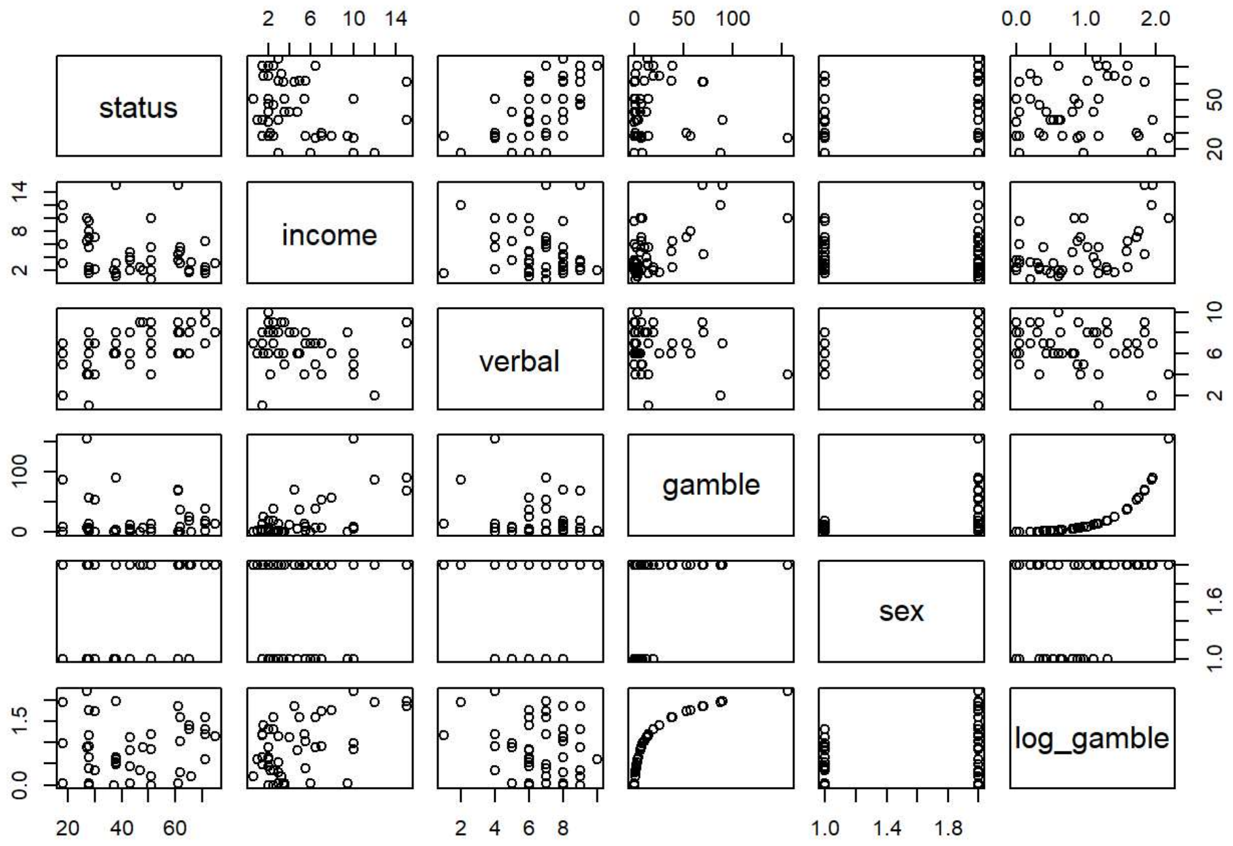


View pairs

Having transformed our variable, let’s use `pairs()` to view the relationsihps between each of the variables.

We can see that `income` may have a linear relationship with `log_gamble`, while `verbal` and `status` are somewhat more muddled.

```
pairs(teengamb)
```

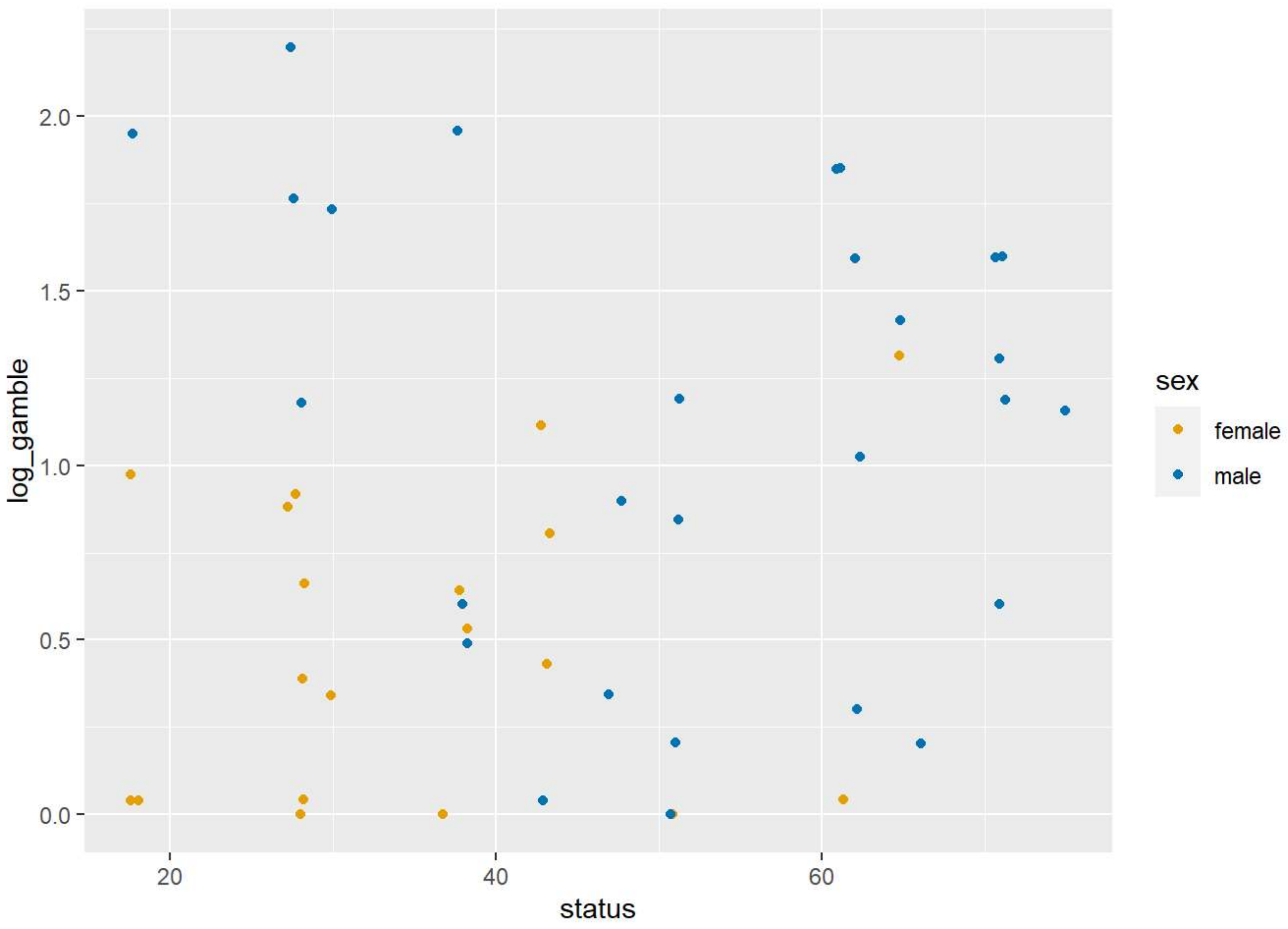


View pairs, view individual relationships separated by sex

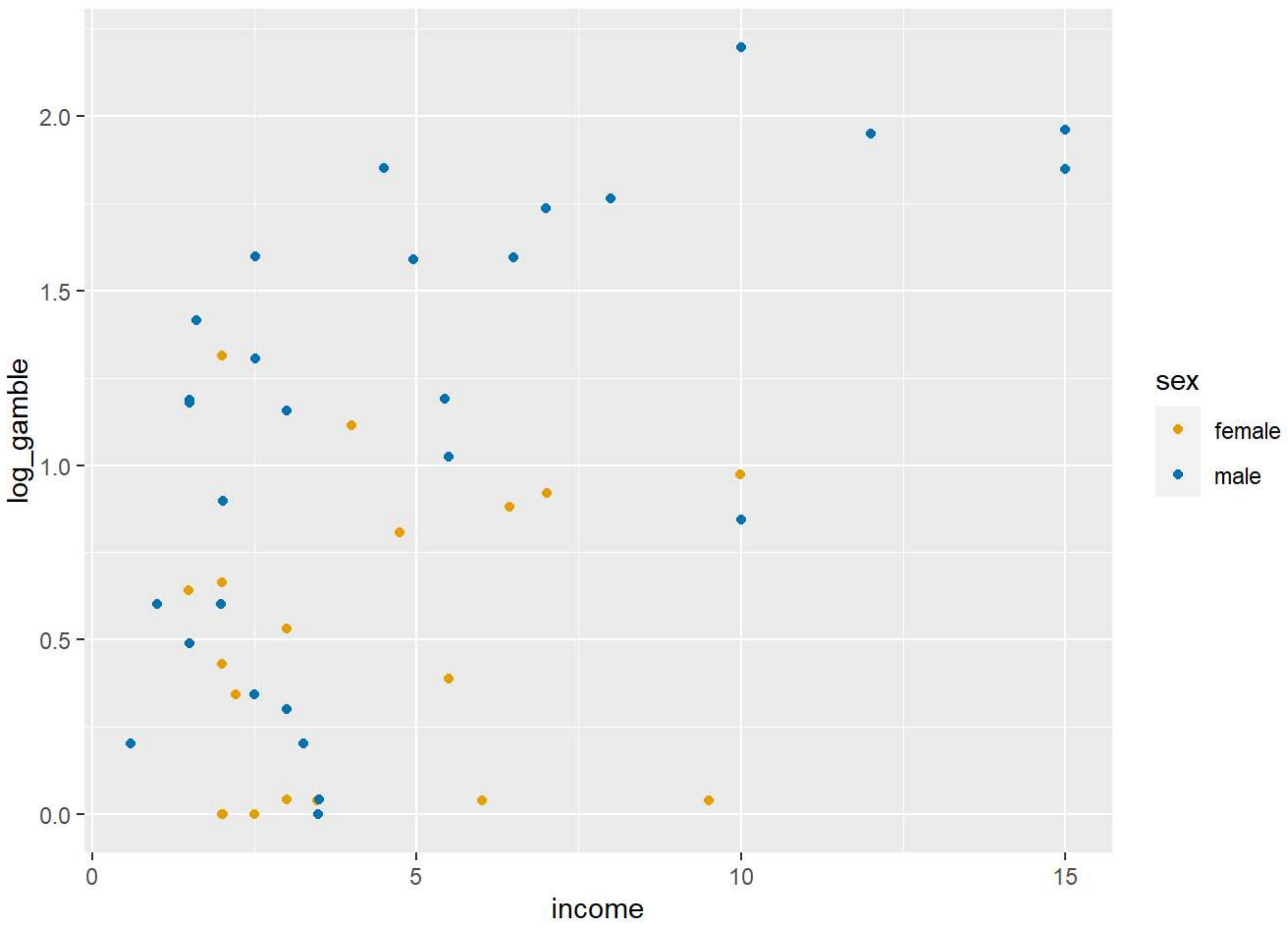
It’s also clear from our `pairs()` output that there’s a noticeable correlation between values with `sex == male` and `log_gamble`.

Let’s view the relationships with our numerical predictors, with a fill color indicating the value of `sex`.

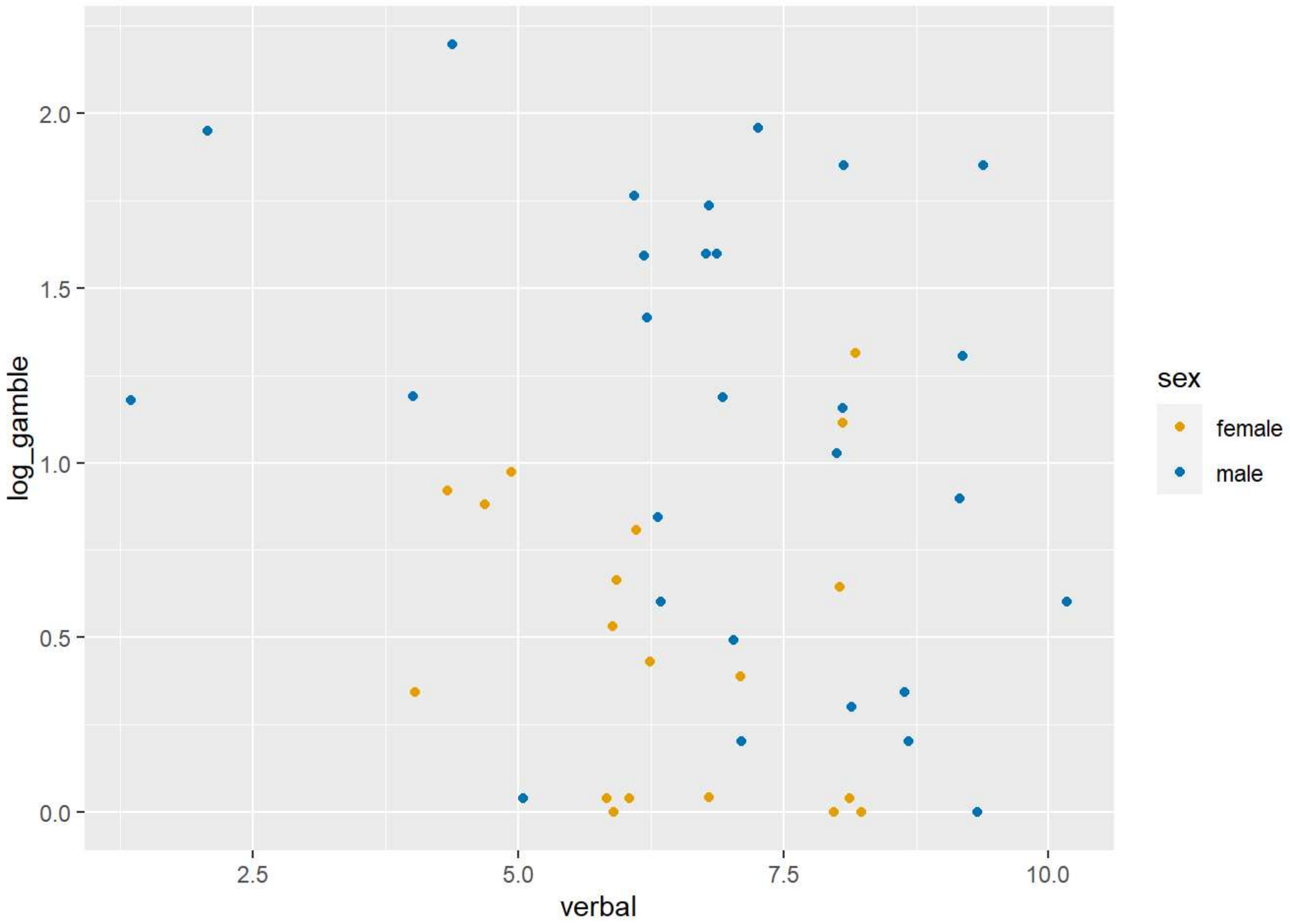
```
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

#This colorblind-friendly color palette is from http://jfly.iam.u-tokyo.ac.jp/color/.  Referencing code courtesy of http://www.cookbook-r.com/Graphs/Colors_(ggplot2)/

#ggplot(data=teengamb, aes(x=status, y=gamble, group=sex, fill = sex)) +
  #geom_jitter(aes(col=sex)) + scale_colour_manual(values=c("#E69F00", "#0072B2"))

ggplot(data=teengamb, aes(x=status, y=log_gamble, group=sex, fill = sex)) + geom_jitter(aes(col=sex)) + scale_colour_manual(values=c("#E69F00", "#0072B2"))
```



```
#ggplot(data=teengamb, aes(x=income, y=gamble, group=sex, fill = sex)) +
  #geom_jitter(aes(col=sex)) + scale_colour_manual(values=c("#E69F00", "#0072B2"))

ggplot(data=teengamb, aes(x=income, y=log_gamble, group=sex, fill = sex)) + geom_jitter(aes(col=
sex)) + scale_colour_manual(values=c("#E69F00", "#0072B2"))
```



```
#ggplot(data=teengamb, aes(x=verbal, y=gamble, group=sex, fill = sex)) +
  #geom_jitter(aes(col=sex)) + scale_colour_manual(values=c("#E69F00", "#0072B2"))

ggplot(data=teengamb, aes(x=verbal, y=log_gamble, group=sex, fill = sex)) + geom_jitter(aes(col=
sex)) + scale_colour_manual(values=c("#E69F00", "#0072B2"))
```



Construct a linear model using all variables

Starting with `sex`, `status`, `income`, and `verbal` as our predictors, we'll fit a linear model to the data.

The distribution of residuals are not symmetrically distributed around their median, and the standard error of the residual is not uniformly close to 1.5 times the two quartiles, so our assumption of normally-distributed residuals is likely violated.

Each of the coefficients are statistically significant at α = 0.05, however it's concerning that the coefficients of `income`, `verbal`, `status`, and `sexmale` are not larger with respect to their respective coefficient standard errors.

We'll want to tweak this model by removing one of the least significant variables.

```
teengamb_lm1_log <- lm(log_gamble ~ sex + status + income + verbal, data=teengamb)
summary(teengamb_lm1_log, cor=TRUE)
```

```
## 
## Call:
## lm(formula = log_gamble ~ sex + status + income + verbal, data = teengamb)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.02065 -0.24696  0.00179  0.31057  0.82655 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.366980   0.306436   1.198   0.2378    
## sexmale      0.378355   0.170539   2.219   0.0320 *  
## status       0.012955   0.005838   2.219   0.0320 *  
## income       0.093654   0.021297   4.398 7.33e-05 ***
## verbal      -0.113631   0.045114  -2.519   0.0157 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4713 on 42 degrees of freedom
## Multiple R-squared:  0.5206, Adjusted R-squared:  0.475 
## F-statistic:  11.4 on 4 and 42 DF,  p-value: 2.347e-06
## 
## Correlation of Coefficients:
##         (Intercept) sexmale status income
## sexmale  0.04                            
## status  -0.27       -0.55                
## income  -0.49       -0.29    0.34        
## verbal  -0.58        0.20   -0.53  -0.02 
```

Construct linear model - backward elimination - remove `status`

Using backward elimination, we'll remove `status` from our model. Our adjusted R^2 has diminished somewhat, but our Residuals are now more normally distributed about their mean.

However, the impact of `verbal` is now no longer statistically significant, so we'll remove it in the next step.

```
teengamb_lm2_log <- lm(log_gamble ~ sex + income + verbal, data=teengamb)
summary(teengamb_lm2_log, cor=TRUE)
```

```
## 
## Call:
## lm(formula = log_gamble ~ sex + income + verbal, data = teengamb)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.06455 -0.33920 -0.00057  0.36434  1.09444 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.55087    0.30818   1.787 0.080909 .  
## sexmale      0.58716    0.14857   3.952 0.000284 ***
## income       0.07781    0.02096   3.712 0.000586 ***
## verbal      -0.06088    0.04005  -1.520 0.135825    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4923 on 43 degrees of freedom
## Multiple R-squared:  0.4645, Adjusted R-squared:  0.4271 
## F-statistic: 12.43 on 3 and 43 DF,  p-value: 5.488e-06
## 
## Correlation of Coefficients:
##         (Intercept) sexmale income
## sexmale -0.13                     
## income  -0.44       -0.14         
## verbal  -0.89       -0.13    0.19 
```

Construct linear model - backward elimination - remove `verbal`

Removing `verbal` from the model yields a new one with symmetrically distributed residuals about their mean, and a residual standard error close to 1.5 times the 1st and 3rd Quartile residuals` absolute distance from the mean.

The two coefficients are statistically significant; however, their values are only 3-4 times their respective standard errors, which means there is likely too much error to make reliable predictions.

This may be a function of too few survey responses.

As a side note, We should also question whether the responses can be considered truly independent of one another, given they were obtained from two classrooms in a school where students knew each other (and likely gambled together), rather than from a representative sample of the population.

```
teengamb_lm3_log <- lm(log_gamble ~ sex + income, data=teengamb)
summary(teengamb_lm3_log, cor=TRUE)
```

```
##
## Call:
## lm(formula = log_gamble ~ sex + income, data = teengamb)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98610 -0.33573  0.03733  0.36488  1.01134
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.13479    0.14367   0.938 0.353298
## sexmale      0.55777    0.14949   3.731 0.000543 ***
## income       0.08387    0.02088   4.017 0.000227 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4996 on 44 degrees of freedom
## Multiple R-squared:  0.4357, Adjusted R-squared:   0.41
## F-statistic: 16.98 on 2 and 44 DF,  p-value: 3.416e-06
##
## Correlation of Coefficients:
##         (Intercept) sexmale
## sexmale -0.54
## income  -0.60       -0.12
```

Remove log transformation and interpret results…

Even though the model is likely too problematic to make reliable predictions, it's worth going through the process of "de-logging" in order to demonstrate interpreting log-transformed results.

In the case of this model, where the response variable has been log-transformed but the predictor variables have not, we'll "de-log" by exponentiating each of the coefficients, subtracting 1, and multiplying by 100 to get the "Percentage impact" on the (untransformed) response variable.

```
lm3_sexmale_est <- teengamb_lm3_log$coefficients[["sexmale"]]
lm3_sexmale_delogged <- (exp(lm3_sexmale_est) - 1)*100

lm3_income_est <- teengamb_lm3_log$coefficients[["income"]]
lm3_income_delogged <- (exp(lm3_income_est) - 1)*100

print(paste0("A value of 'male' for variable `sex` increases the value of `gamble` by ", round(lm3_sexmale_delogged, 3), "%."))
```

```
## [1] "A value of 'male' for variable `sex` increases the value of `gamble` by 74.678%."
```

```
cat("\n")
```

```
print(paste0("Each 1 GBP increase in `income` increases the value of `gamble` by ", round(lm3_income_delogged, 3), "%."))
```

```
## [1] "Each 1 GBP increase in `income` increases the value of `gamble` by 8.749%."
```