CODE EXAMPLE - Randomization Tests (Stat 230 - c4.6)

P.B.Matheson adapted from S.M. Liao & A.S. Wagaman

Inference for linear regression only holds if the appropriate conditions (parameters) are met. We know how to assess whether these conditions are met, and we have learned a few techniques for correcting them when they are not (e.g. re-expressions).

Now we will explore randomization as a technique for making *non-parametric* inferences. Such inferences do not rely on stringent assumptions about the distribution of the error terms (normally distributed and equal variance).

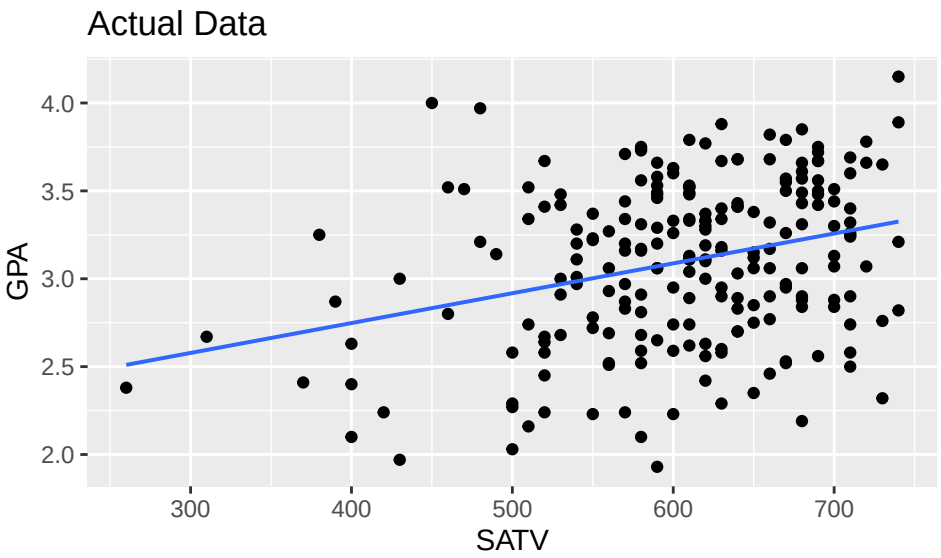
In this example, suppose we are trying to explain a college student's GPA as a function of their score on the verbal section of the SAT (i.e. using an SLR).

```
data(FirstYearGPA)
# Use only the non-null entries
FirstYearGPA <- subset(FirstYearGPA, !is.na(GPA))
# How big is our sample?
nrow(FirstYearGPA)
```

```
## [1] 219
```

First, let's examine the relationship between these two variables graphically.

```
gf_point(GPA ~ SATV, data = FirstYearGPA, title = "Actual Data") %>%
  gf_lm()
```



There appears to be some linear association between these variables, but it is not particularly strong. We can quantify this relationship using the correlation coefficient. We will put the actual correlation coefficient into `cor.actual` so we can access it later.

```
cor.actual <- cor(GPA ~ SATV, data = FirstYearGPA)
cor.actual
```

```
## [1] 0.304311
```

The value of the correlation coefficient is about 0.3, which is not large, but does appear to be significantly different from 0. We can use cor.test to evaluate whether or not this r is significantly different from zero.

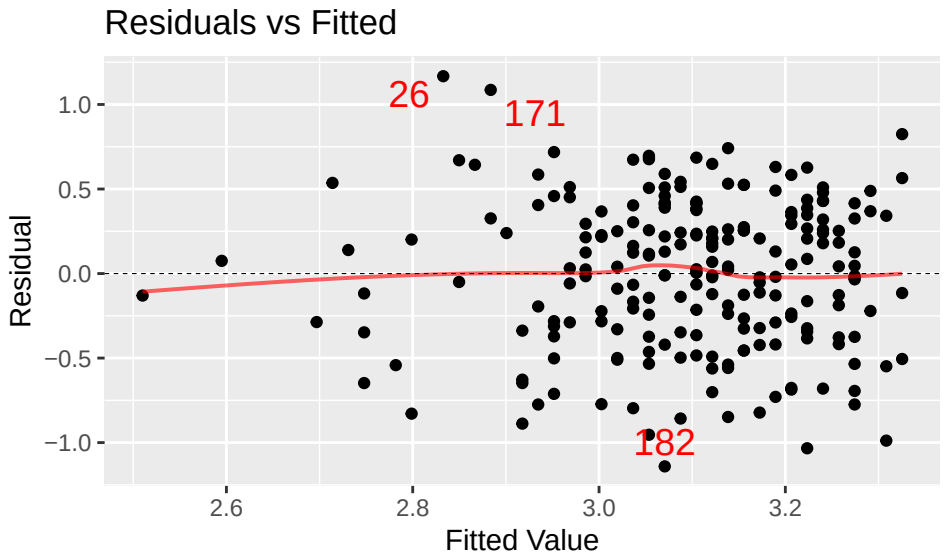
```
cor.test(GPA ~ SATV, data = FirstYearGPA) #test to check for non-zero correlation
```

```
##
##  Pearson's product-moment correlation
##
## data:  GPA and SATV
## t = 4.706, df = 217, p-value = 4.5e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.178958 0.419943
## sample estimates:
##      cor
## 0.304311
```

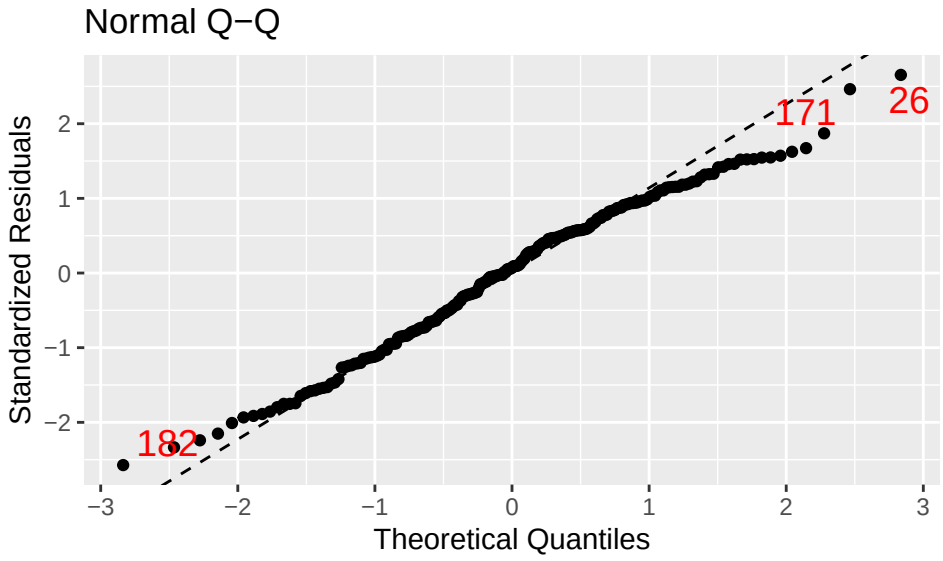
This test confirms that the correlation between college GPA and $SATV$ is most likely not zero, but the validity of this test requires the assumptions for simple linear regression to be met (in particular, the normality condition on the errors MUST hold). Let's check that here.

```
mod <- lm(GPA ~ SATV, data = FirstYearGPA)
mplot(mod, which = 1)
```

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




```
mplot(mod, which = 2)
```



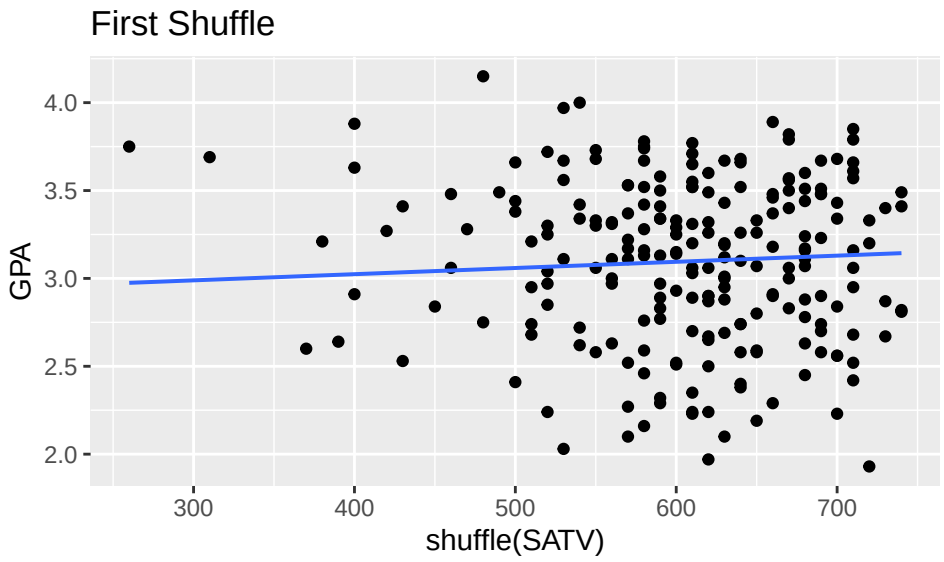
Using the randomization test to see if our correlation coefficient is significant (SLR example)

If GPA and $SATV$ were really correlated, then there is a real relationship binding the i^{th} value of GPA to the i^{th} value of $SATV$. In this case, it would not make sense to link the i^{th} value of GPA to the some other value of $SATV$. But if the correlation between these two variables was in fact zero, then it wouldn't matter how we matched up the entries in the variables!

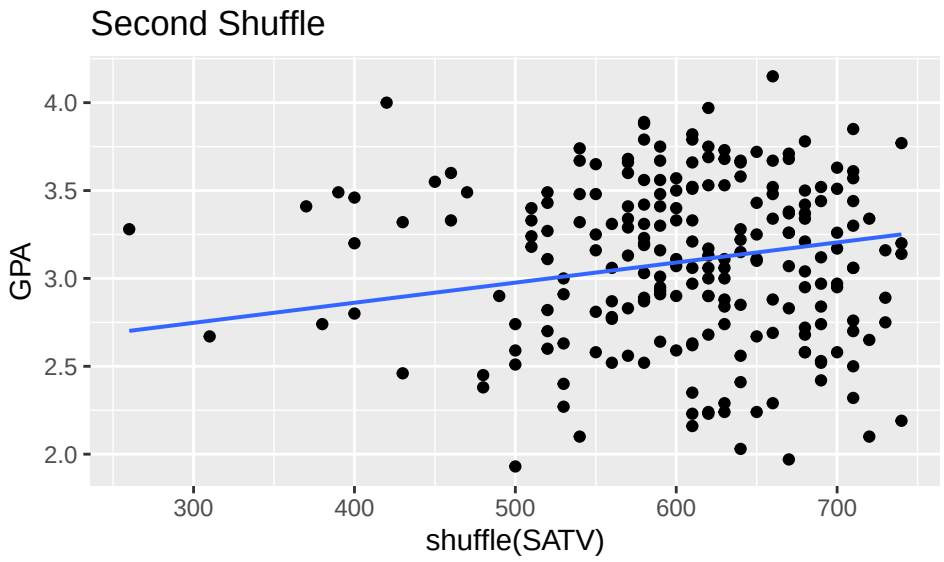
The basic idea of a randomization or permutation test is to shuffle the mapping between the two variables (or just the values of one variable) many times and examine the distribution of the resulting correlation coefficient (or whatever statistic is your focus and is appropriate for the situation). If the actual value of the correlation coefficient you observed between the real x and y data is a rare member of that distribution, then we have evidence that the true correlation is non-zero.

R can shuffle the data for us, and we can watch what happens to the relationship between Y and our shuffled x by looking at regression lines after each round of shuffling. We can also record summary statistics - like the correlation, to see how it behaves when we are "destroying" the relationships between the variables.

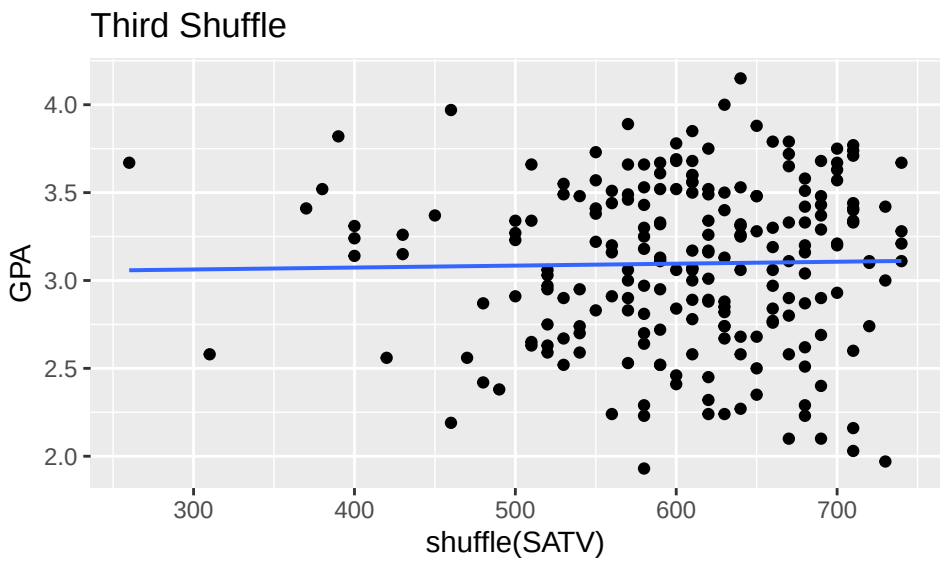
```
gf_point(GPA ~ shuffle(SATV), data = FirstYearGPA, title = "First Shuffle") %>%
  gf_lm()
```



```
gf_point(GPA ~ shuffle(SATV), data = FirstYearGPA, title = "Second Shuffle") %>%
  gf_lm()
```




```
gf_point(GPA ~ shuffle(SATV), data = FirstYearGPA, title = "Third Shuffle") %>%
  gf_lm()
```



Note that we are permuting (shuffling) the predictor. In some settings, you might want to permute the response - it depends on what null distribution you are aiming for (for example, to test if the overall model is useful). Most of our examples will involving permuting the predictor (looking at whether that predictor is significant).

The procedure for the randomization test in this instance is simple. We simply shuffle the explanatory variable and compute the resulting correlation coefficient with the response variable. The reason we shuffle the predictor of interest is that you want each case to still have the same relationships between the response and OTHER predictors, so you just shuffle the one predictor you want to learn about.

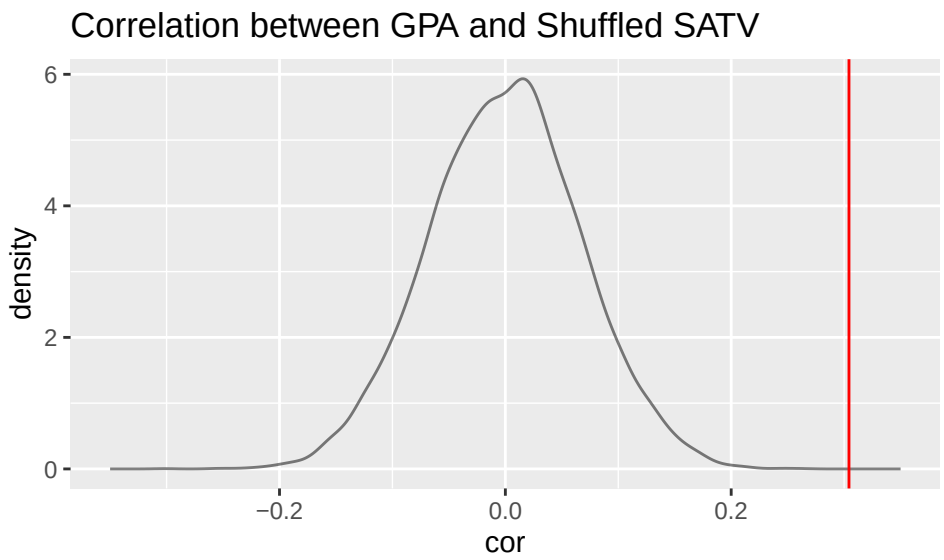
We do this many times, until we have a sense of the distribution of that shuffled correlation coefficient would look like when there is no relationship between x and y. We then examine where the observed correlation falls in that distribution.

NOTE: Randomization can be used to look at lots of different statistics (means, differences between means, etc.). In this example, we are just focused on a correlation coefficient but the methods can be applie in many settings.

```
#set the seed so we get same results when using this example in class
set.seed(230)
# Do this 10000 times, it will take a couple of minutes
rtest <- do(10000) * cor(GPA ~ shuffle(SATV), use = "pairwise.complete.obs", data = FirstYearGPA)
names(rtest) #look to see what are the objects were saved - here it is only cor
```

```
## [1] "cor"
```

```
gf_dens(~ cor, data = rtest) %>%
  gf_lims(x = c(-0.35, 0.35)) %>%
  gf_labs(title = "Correlation between GPA and Shuffled SATV") %>%
  gf_vline(xintercept = ~ cor.actual, color = "red")
```



We can also explicitly find where in the distribution the observed correlation lies using the command pdata.

```
#provides area below cor.actual based on rtest$result distribution
pdata(~ cor, cor.actual, data = rtest)
```

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

```
#area above (if you want upper tail - LOOK at the distribution)
pdata(~ cor, cor.actual, data = rtest, lower.tail = FALSE)
```

```
## [1] 0
```

```
## Equivalently, we can use `pdata()` function to calculate such a p-value
pdata(rtest$cor, abs(cor.actual), lower.tail=FALSE) +
  pdata(rtest$cor, -abs(cor.actual))
```

```
## [1] 0
```

The latter version gives us the exact probability of getting a correlation coefficient like the one we observed (cor.actual) if the null is true (rho=0, aka there is no relationship between GPA and SATV).

Finally, using the command qdata, we can use the quantiles fot the middle 95% of the distribution (2.5% and 97.5%) and create a nonparametric 95% confidence interval for the correlation coefficient. Note that this is a bit different from our usual CIs where we find the t distribution cutoff values. Instead of giving us a range of

reasonable values for the correlation, this interval gives you a range of values for the correlation that you'd observe if there was NO relationship between the variables.

The interpretation here is that if our actual correlation value was contained in this interval, we would NOT consider it to be statistically significant. This interval shows the likely range of correlation coefficients that would come from the set of correlations we computed when the relationship between variables was destroyed.

```
qdata(~ cor, c(0.025, 0.975), data = rtest)
```

```
##      2.5%     97.5%
## -0.131584  0.131511
```

Since our actual correlation of 0.3 lies entirely above this interval, we believe we have a statistically significant result. In other words, 0.3 is not likely to happen by chance alone when there is, in reality, no relationship between SATV and GPA (due to the shuffling).

Randomization Test in MLR (for the slope coefficients)

Let's suppose that we still want to assess if `SATV` has a significant relationship with `GPA` (as a predictor) when `HSGPA` and `FirstGen` are also included in the model.

Let's fit the model first.

```
m2 <- lm(GPA ~ SATV + HSGPA + FirstGen, data = FirstYearGPA)
msummary(m2)
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.715988   0.295003    2.43    0.016 *
## SATV         0.001012   0.000348    2.91    0.004 **
## HSGPA        0.518540   0.074956    6.92  5.2e-11 ***
## FirstGen    -0.199837   0.089154   -2.24    0.026 *
##
## Residual standard error: 0.402 on 215 degrees of freedom
## Multiple R-squared:  0.263,  Adjusted R-squared:  0.253
## F-statistic: 25.6 on 3 and 215 DF,  p-value: 3.35e-14
```

```
m2$coefficients["SATV"]    #extract the estimated coeff. for `SATV`
```

```
##       SATV
## 0.00101246
```

```
#mplot(m2, which = 1)
#mplot(m2, which = 2)
```

Note that estimated coefficient for `SATV` is about 0.001. From the t-test, is `SATV` significant? Again, we'd need to believe the regression **conditions** are met in order to proceed with this inference.

We can assess the significance of the coefficient with a randomization test. We want to leave the relationships between `GPA` and the other predictors alone, and only shuffle the values of `SATV`, the predictor of interest. We could then obtain coefficients for `SATV` in the shuffled models, and see how those *simulated* coefficients (simulated under H_0) compare to the actual one we get for the non-shuffled model (i.e. 0.001).

```
set.seed(0918)
permtest2 <- do(10000)*lm(GPA ~ shuffle(SATV) + HSGPA + FirstGen, data = FirstYearGPA)
names(permtest2)
```

```
##  [1] "Intercept" "SATV"      "HSGPA"     "FirstGen"  "sigma"     "r.squared"
##  [7] "F"         "numdf"     "dendf"     ".row"      ".index"
```

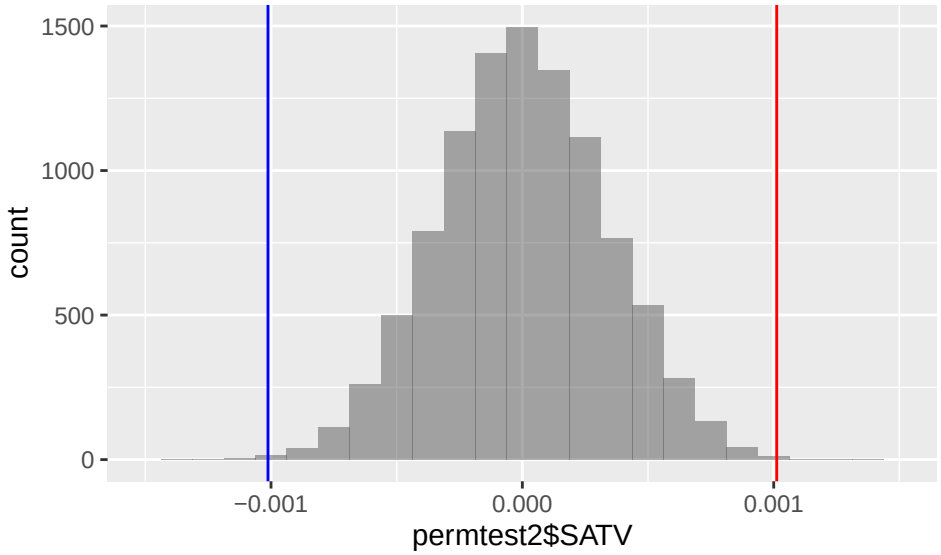
Here you can see that R saves all coefficients (including the intercept), residual standard error (named as sigma), r-squared values ("r.squared), and others for you.

Let's see if the coefficient of SATV was really significant in this model by checking if the actual coefficient estimate of SATV (0.00101) is really unusual/extreme under the simulated null distribution.

```
pdata(permtest2$SATV, 0.001012, lower.tail=FALSE) + pdata(permtest2$SATV, -0.001012)
```

```
## [1] 0.0017
```

```
gf_histogram(~permtest2$SATV) %>%
  gf_lims(x = c(-0.0015, 0.0015)) %>%                    ## specify the range on the x-axis
  gf_vline(xintercept = ~ 0.001012, color = "red") %>%  ## plot a red vertical line at 0.001012
  gf_vline(xintercept = ~ -0.001012, color = "blue")    ## plot a blue vertical line at -0.001012
```



The permutation p-value is 0.002, indicating that it's very unlikely for us to get a coefficient as extreme as 0.00101 for SATV if there was no relationship between SATV and GPA with HSGPA and FirstGen in the model. In other words, it appears that SATV is a *significant* predictor for GPA when HSGPA and FirstGen are in the model.

Finally, we can also find a **permutation 95% confidence interval** for this coefficient. The interpretation here is that **coefficient values IN this interval would NOT be considered statistically significant**, because this interval is coming from the set of coefficients we computed when the relationship between SATV and GPA were destroyed (i.e. under $H_0 : \beta_{SATV} = 0$) with HSGPA and FirstGen in the model.

```
qdata(permtest2$SATV, c(0.025, 0.975))
```

```
##          2.5%         97.5%
## -0.000640727  0.000655500
```

Our actual coefficient for SATV is 0.00101, which lies completely above this interval, so on the basis of this randomization test we believe we have a statistically significant result here.