# FDA Spring Semester Summary

Peter Norwood

May 2018

## Getting Oriented

Last semester Jacob Alfieri and I set groundwork for what continued this semester. We familiarized ourselves with functional data, quantile regression, and Dr. Arnab Maity's proposed KMQR method. We generated simulated data to test the method on and found promising results. Our future work revolved around mixing the linear kernel and the quadratic kernel into one test, rather than making the kernel a parameter for the test.

Note: some of this material, particularly the methods to analyze power, is the same as last semester. The difference is the new method, with the mixed kernel, is being tested.

## Mixed KMQR

In the fall semester, our KMQR function either assumed a linear function or a quadratic function. The kernel parameter was set to 1 or 2 and we found when paired with the appropriate function (kernel = 1 with linear function, kernel = 2 with quadratic function), KMQR had promising results. The new, mixed KMQR, test goes as follows:

1. Let $a =$ the p-value from the linear kernel test, $b =$ the p-value from the quadratic kernel test.
2. Let $"tobs" = ln(a) + ln(b)$. This is our test statistic and we need test it to the null distribtuion, which is estimated through a bootstrap.
3. Randomly permute $y$, the response variable, and run the two tests again. The same covariates are used in the test, only the response is permuted. Let $a* =$ the p-value from the linear test, $b* =$ the p-value from the quadratic test. For each permutation, save the $ln(a*) + ln(b*)$. Repeat this process 500 times, ending with a vector of 500 $ln(a*) + ln(b*)$, which is the simulated distribution of p-values.
4. Then your final p-value is the percentage of instances where our test statistic is greater than a distribution value. It is calculated by:

$$p = \frac{(\sum_{i=1}^{500} r_i) + 1}{500 + 1},$$

where:

$$r_i = \begin{cases} 1, & \text{if } r_i < tobs \\ 0, & \text{otherwise} \end{cases}$$

The idea for this method originally came from *The potential for increased power from combining P-values testing the same hypothesis*, Gaju & Ma (2017).

# Methods

Once the Mixed KMQR Function was written, it was tested using the same simulated data method from last semester. This helped answer the research question: does the the Mixed KMQR method to analyze function data provide meaningful results?

The functional curves were simulated as follows for 51 equally spaced points of $t$, where $t \in [0, 1]$:

$$x_i(t) = x_{i,1} + x_{i,2}\cos(2\pi t) + x_{i,3}\sin(2\pi t) + e_{i,t}N(0,1)$$

where:

$$x_{i,1} \sim N(0,4)$$
$$x_{i,2} \sim N(0,2)$$
$$x_{i,3} \sim N(0,1)$$

The response variable was simulated as follows:

**$\mathbf{X_2}$ Method**

$$\begin{aligned} y_i &= 1 + \delta\left[\int_0^1 x_i(t)\cos(2\pi t)dt\right]^k + N(0,1) \\ &= 1 + \delta[x_{i,2}]^k + N(0,1) \end{aligned}$$

**Average Method**

This is intended to provide a stronger signal than the $X_2$ method.

$$y_i = 1 + \delta\left[\frac{x_{i,1} + x_{i,2} + x_{i,3}}{3}\right]^k + N(0,1)$$

where:
$\delta$: determines the strength of the relationship between the functional data and the response variable
$k$: determines the nature of the relationship between the functional data and the response variable (linear, quadratic, etc.)

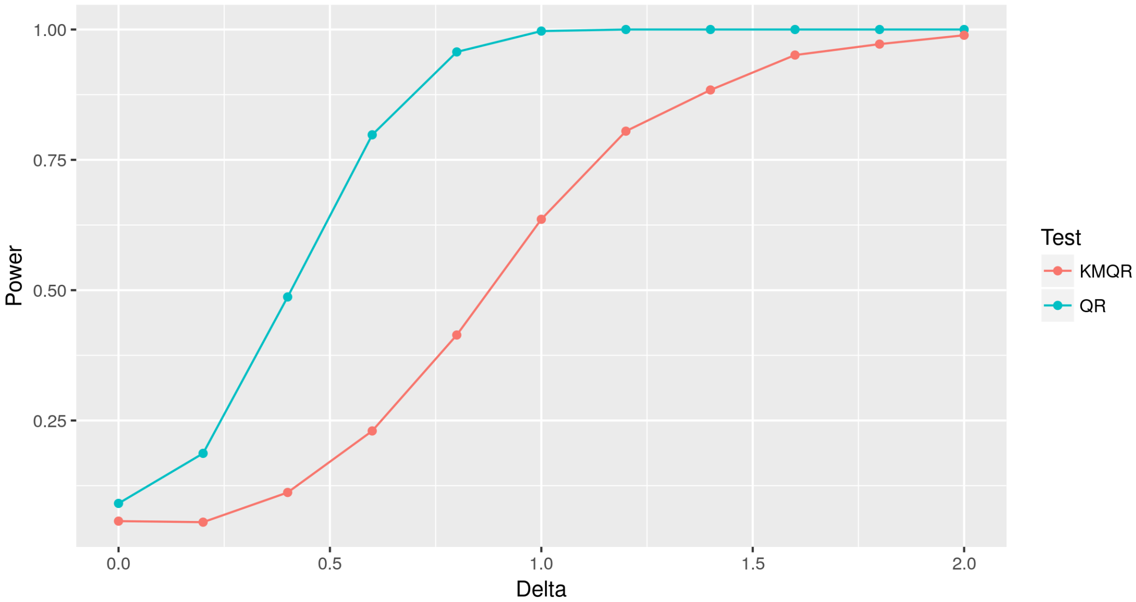
## Simulations

Using the $X_2$ and Average methods to simulate data, we compared the power and type I error of the Mixed KMQR method to the basic QR technique across 11 different choices of $\delta \in [0, 2]$, when the integral power, $k = 1$. For each value of $\delta$ we simulated 10,000 datasets of 100 functional curves. To investigate sample size's role, we repeated the process but for 10,00 datasets of 200 functional curves. We calculated power as the percent of simulations where an ANOVA test rejected the null model, which only included an intercept and not the functional curves, at $\alpha = .05$.

# Results

## Comparing KMQR and QR when k = 1

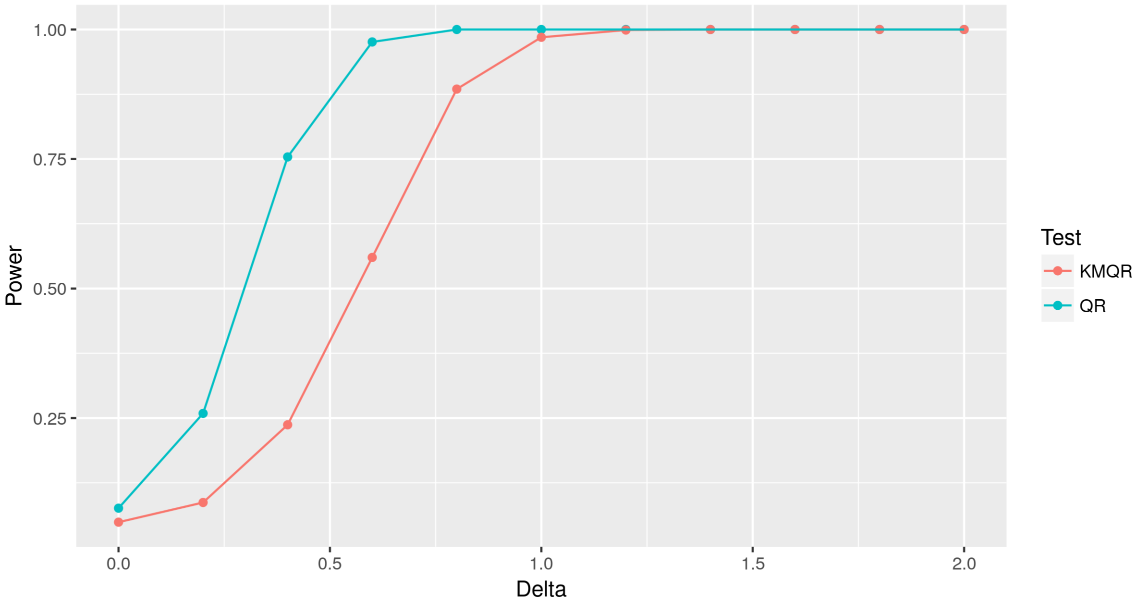
Simulations used the $X_2$ method, k=1 and a sample size of 100.



As the chart shows, we found that the QR method had a higher power across the different $\delta$ values. However, we also found that the level of significance for the QR test was not actually .05. Instead, the type I error rate was about .09. Since the QR test does not attain the desired level of significance, the power is troublesome.

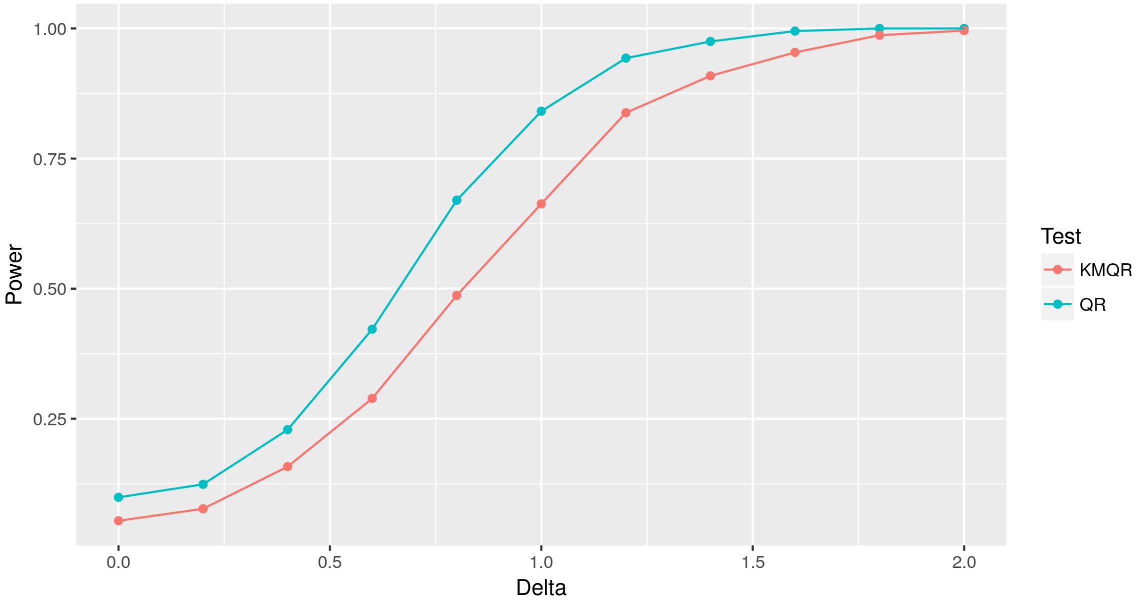
We found similar results when we tested 10,000 datasets of 200 functional curves, with both the QR and Mixed KMQR performing better.

Simulations used the $X_2$ method, k=1 and a sample size of 200.

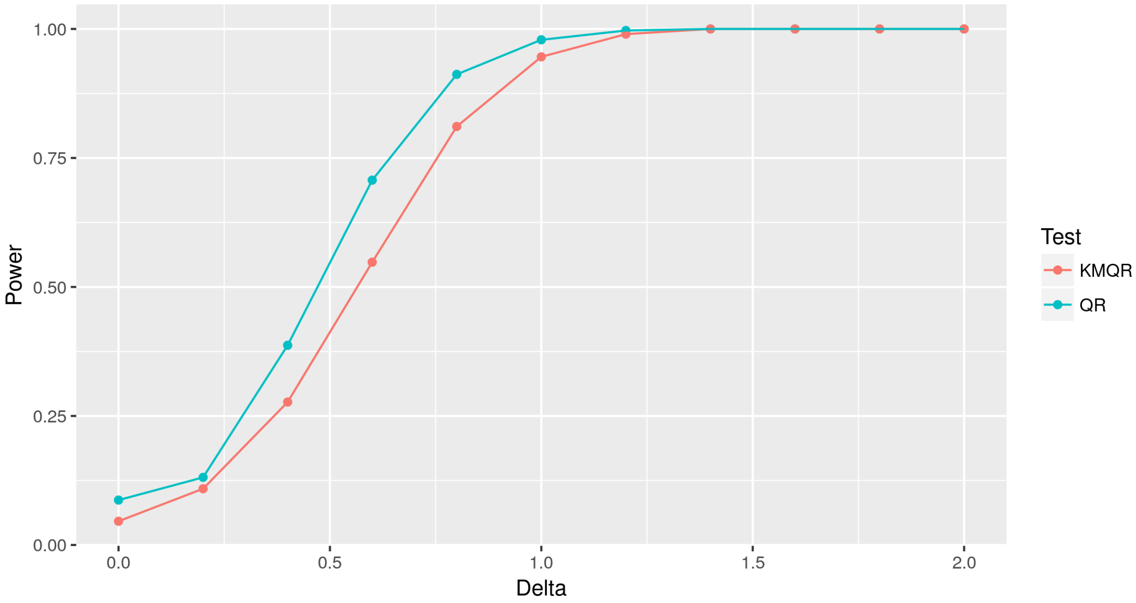


Similar results are found with the Average method as well.

Simulations used the Average method, k=1 and a sample size of 100.



Simulations used the Average method, k=1 and a sample size of 200.

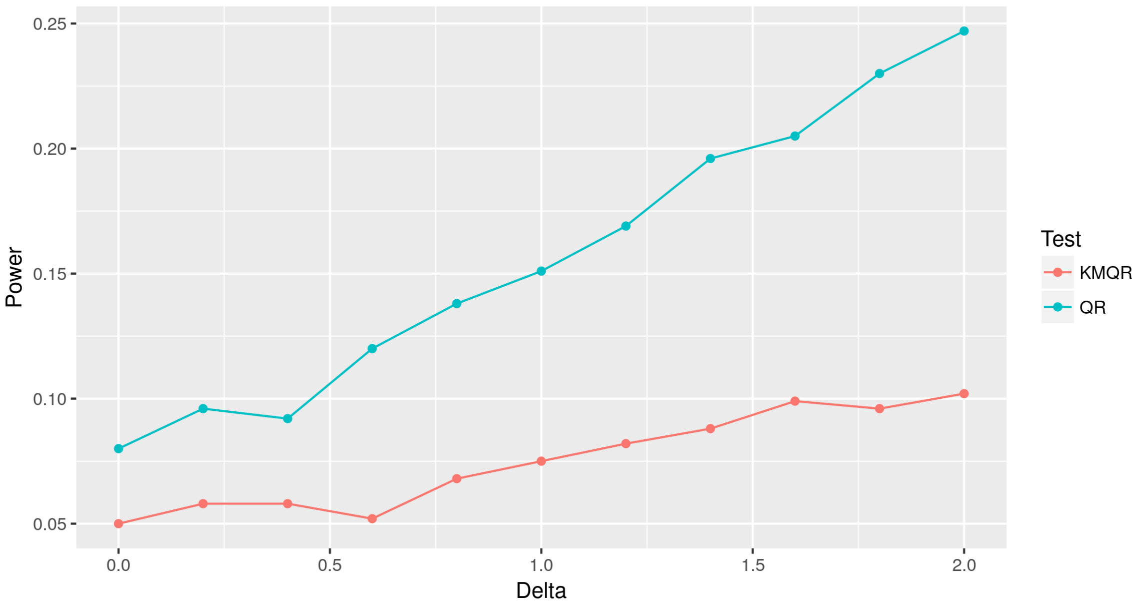


## Comparing KMQR and QR when k = 2

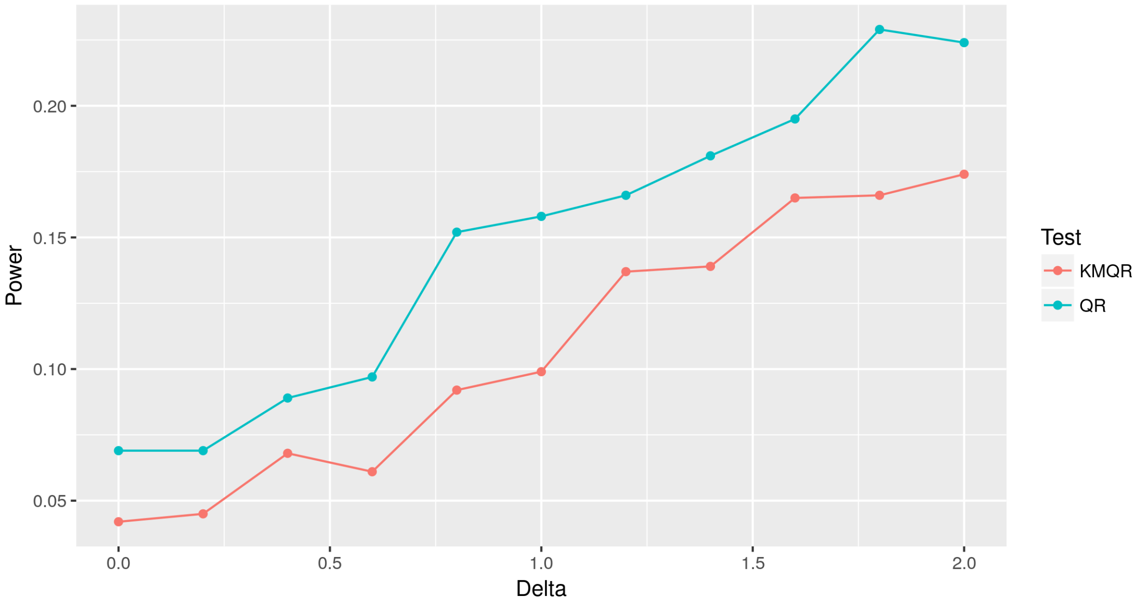
The Mixed KMQR performed fairly well when tested on a linear function and had the appropriate type I error. The main area of investigation, however, is how does KMQR perform on quadratic data. Thus, we changed our simulated data from a linear function to a quadratic one, and ran the same tests.

Initially, with the $X_2$ Method, the results were not promising. Both the QR and Mixed KMQR had dismal results.

Simulations used the $X_2$ method, k=2 and a sample size of 100.

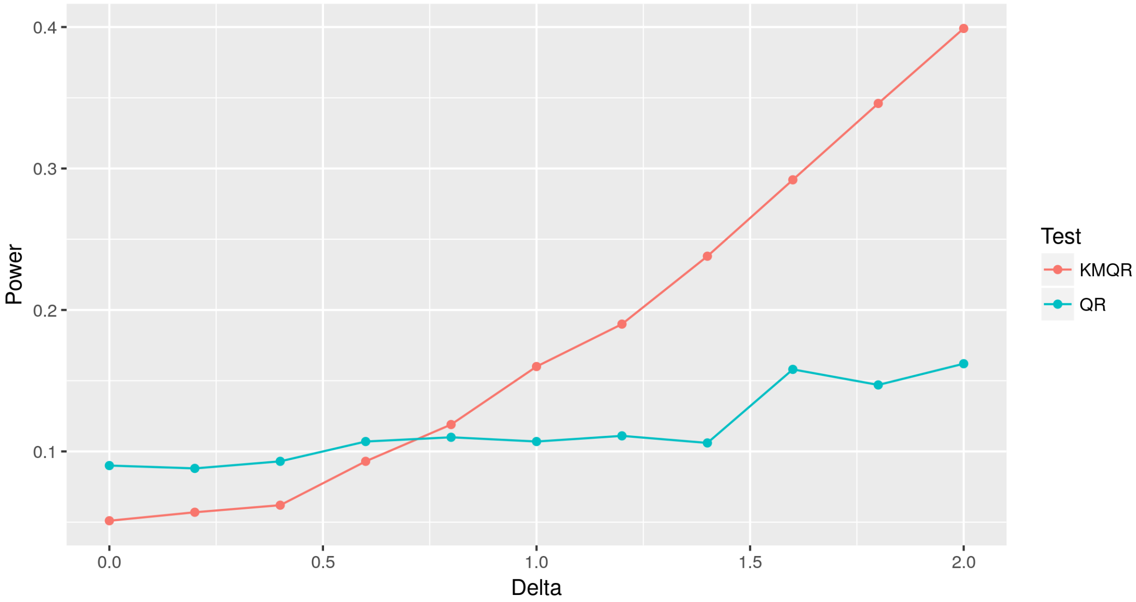


Simulations used the $X_2$ method, k=2 and a sample size of 200

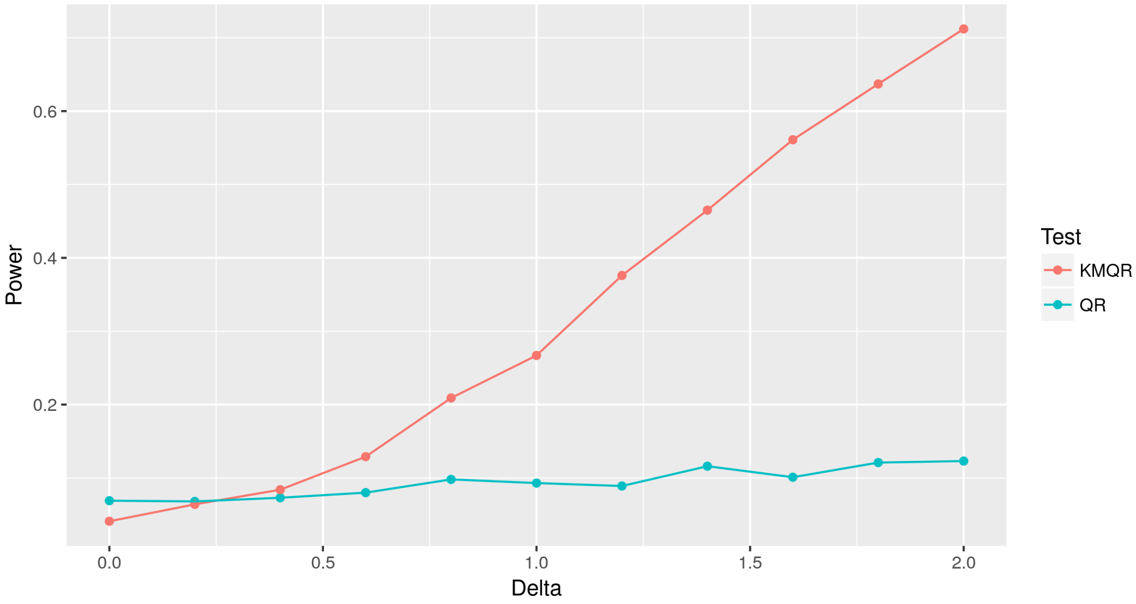


For the average method, more encouraging results were found. While still not to the level of a linear function, the Mixed KMQR outperforms the QR function.

Simulations used the Average method, k=2 and a sample size of 100.



Simulations used the Average method, k=2 and a sample size of 200



**Power at $\delta = 2$ for Simulations Using $X_2$**

| Simulation Exponent ($k$) | Power for $n = 100$ | Power for $n = 200$ |
|---|---|---|
| 1 | .989 | $>$ .999 |
| 2 | .102 | .174 |

**Power at $\delta = 2$ for Simulations Using Average of $\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}$**

| Simulation Exponent ($k$) | Power for $n = 100$ | Power for $n = 200$ |
|---|---|---|
| 1 | .996 | $>$ .999 |
| 2 | .399 | .712 |

# Discussion

Overall, the Mixed KMQR method showed potential. The type I error was at .05 and it outperforms the standard QR method when modeling a quadratic function.

When modeling the $X_2$ method on a linear function, Mixed KMQR performed similarly to standard QR, but with the correct type I error. For the $X_2$ method on a quadratic function, both Mixed KMQR and QR performed poorly.

The Average method was similar to the $X_2$ for linear functions. Both Mixed KMQR and QR had solid looking power and Mixed KMQR had the correct type I error. When modeling a quadratic function, QR was as dismal as the $X_2$ method, but Mixed KMQR showed promise. The power is still underwhelming compared to the linear functions, but the power curves are much better than QR.

# Future Work

Considerable work is still needed to improve and evaluate the Mixed KMQR method. Dr. Maity and I believe four items are the next steps:

1. Adding Additional Kernels

   Currently Mixed KMQR only uses a linear and quadratic KMQR methods and tries to understand the distribution of p-values from those kernels. Since not all data behave linearly or quadratically, additional additional kernels could help to model more behaviors. The main kernel of interest is a Gaussian kernel and a Cubic kernel could also be useful.

2. Scaling PCA Scores

   The poor performance of $X_2$ method with a quadratic function and the better performance of the average method have sparked the idea to scale the PCA scores. We believe the $X_2$ covariate may be overwhelmed by the other covariates and does not provide enough signal. By dividing the PCA scores by their standard deviations, we could standardize them and have allow $X_2$ to have more signal.

3. Increasing Computing Speed

   Inside of the Mixed KMQR, the response variable is permuted 500 times to try to produce a distribution of the p-values. As one can imagine, this requires a significant amount of time, especially when testing the functions. Currently all of the function's code is written in R and optimized to an extent. To continue to speed up Mixed KMQR, the code needs to be translated to a faster computing language. One way to do this is the Rcpp package in R, which allows one to write C++ code in an R script. More information can be found at Rcpp's CRAN page.

4. Expand the Bootstrap

   After a quicker function has been written, increasing the bootstrapped sample could help to better understand the distribution of the p-values. Permuting the response 1000 or 5000 times and then retesting Mixed KMQR could provide interesting results.

# References

Ganju, J & Ma, G, 2017, ”The potential for increased power from combining P-values testing the same hypothesis”, Statistical Methods in Medical Research, vol. 26, no. 1, pp. 64-74.