Master Tutorial

TITLE

Natural Language Processing with R and Python

SHORTENED TITLE

NLP with R and Python

CITATION

Nydick, S. W., Wiseman, B., & Wisner, T. (2022). Natural language processing with R and Python [Master Tutorial]. Society for Industrial and Organizational Psychology Annual Conference, Seattle, WA, United States.

ABSTRACT

Psychologists and statisticians prefer R for data processing and analysis. Data scientists and machine learning practitioners often use Python for text processing. Many of the Python language processing tools are more advanced and developed than those in R. This tutorial session will show you how to integrate Python natural language processing tools within an R workflow. Bring your laptop for this interactive session.

PRESS PARAGRAPH

Statisticians and data scientists often treat R and Python as mutually incompatible tools. Each language has a separate fanbase who are convinced of their language’s superiority: Python is a better general purpose language with modules for neural networks (such as Tensorflow and PyTorch), machine learning (such as scikit-learn), and API creation (such as Flask and Django); R is better for data exploration with packages for data processing (such as dplyr and data.table) and graphing (such as ggplot2). In reality, both R and Python can be used together and without ever leaving the R console. This session will provide a hands-on tutorial for how to integrate R and Python when processing text-based data.

WORD COUNT: ####

**Natural Language Processing with R and Python**

R (R Core Team, 2021) is an open-source programming language that is designed for statistical computing (Hornik, 2017). R can perform anything from standard data analysis to machine learning and natural language processing to highly specialized computations that may be unique to a scientific field. R is not just a statistical analysis package, but a fully-fledged programming *language*. R even managed to become (and remains) one of the ten most popular programming languages (Cass, 2018; Cass, 2021) partly due to the large ecosystem of support pages, books, blogs, tutorials, and R specific conferences. Several R packages, including dplyr (Wickham, François, Henry, and Müller, 2019), ggplot2 (Wickham, 2016), and data.table (Dowle and Srinivasan, 2019) have over 12,000 mentions on Stack Overflow (Robinson, 2017).

Python (Python Software Foundation, 2021) is a general purpose scripting language often used to design websites, pull information from APIs, and process/analyze data. Recently, a large ecosystem of modules have been developed to read and process data (e.g., Harris, Millman, van der Walt, et al., 2020; The pandas development team, 2021), perform basic machine learning algorithms (e.g., Pedregosa et. Al., 2011), and run computationally intensive neural networks (e.g., Abadi et al., 2015; Chollet et al., 2015; Paszke et al., 2019).

Software developers, data scientists, and I/O practitioners can contribute to each language by writing new and unique software, called “packages” in R and “modules” or “libraries” in Python. At the time of this writing, the Comprehensive R Archive Network (CRAN) contains 18,093 available packages, including packages to read data in varying formats (e.g., readr, open.xlsx, haven, rjson, officer, vroom), access databases (e.g., DBI, odbc, RSQLite), clean data (e.g., dplyr, tidyr, stringr, reshape2), perform data analyses and machine learning (e.g., infer, caret, xgboost, randomForest, survival, DALEX), visualize results (e.g., shiny, ggplot2), and interface with other programming languages (e.g., Rcpp, reticulate, RJava). Python libraries are a bit more difficult to enumerate due to a more distributed set of repositories (including PyPI, CloudRepo, Anaconda, among others). However, the public python repository PyPI contains over 324,836 libraries. Python has vastly more libraries than R (approximately 300 thousand versus approximately 20 thousand) due to the wider audience of the Python language as well as the stricter submission guidelines (as well as the continual maintenance by the R Core Team) of R packages. Simply put, it is much harder to publish a package on CRAN than PyPI because each R package needs to pass a set of strict checks on a variety of operating systems.

This Master Tutorial will teach attendees how to run Python modules within an R session/workflow for the purpose of analyzing text data. We will show users how to process text data with R, send R objects to Python for analyses, and return objects back to Python for displaying and reporting results. One can think of this Master Tutorial as a direct continuation of the R Master Tutorials from the 2019 conference on web scraping and the 2021 conference on text analytics with R: if R does not provide enough tools for analyzing data, how can you take advantage of more robust machine learning/NLP tools within an R workflow? Attendees should be familiar with R and have both R and RStudio installed prior to the workshop. We will walk through and explain each line of code in detail, but we will have little time to review the basics of R itself. Note that we will provide a brief overview of Python as well as how to install Python via the R package “reticulate”.

*Proposed Session*

Although psychologists are adept at creating simple multiple choice or Likert-based questionnaires, a lot of information is still contained within unstructured text data. From performance reviews to company ratings, sentiment analyses and matching algorithms can power purchase recommendations, job suggestions, and promotion decisions. Moreover, how people feel about an organization depends not just on simple ratings but on what they say and how they communicate. Just as a picture is worth a thousand words, a sentence is worth a thousand ratings. However, cleaning and parsing text data requires special strategies. R has many tools designed to clean, describe, and display text data (e.g., Munzert, Rubba, Meißner, and Nyhuis, 2015; Silge and Robinson, 2017). However, many of the most commonly used tools for modeling, classifying, and predicting outcomes using text data were built in Python (e.g., Abadi et al., 2015; Chollet et al., 2015; Paszke et al., 2019). Moreover, online courses for natural language processing (e.g., Mourri, Kaiser, & Shyu, n.d.) typically teach only Python tools for text processing. Ideally, one could clean and process data using R but still take advantage of the powerful models in Python. Until recently, tools linking Python to R were ad hoc and required fragile system setups (such as rPython to run Python from R and rpy/rpy2 to run R from Python). More recently, the “reticulate” package was released to better facilitate specifying, updating, and integrating Python within an R environment (Ushey, Allaire, & Tang, 2021). Unfortunately, I/O psychologists often lack the skills required to analyze text data in Python and integrate Python tools within an R workflow. Building on several R-based master tutorials over the last few years (e.g., Jones, Nydick, & Wiseman, 2019a; Jones, Nydick, & Wiseman, 2019b; Jones, Nydick, & Wiseman, 2021a; Jones, Nydick, & Wiseman, 2021b), the proposed tutorial aims to break down useful R (and now Python!) methods for I/O psychologists. Specifically, this tutorial can be seen as an extension of the Web Scraping and APIs with R (Jones, Nydick, & Wiseman, 2019b) and Text Analytics and NLP with R (Jones, Nydick, & Wiseman, 2021b) tutorials that describes how to exploit state-of-the-art methods for understanding text data pulled from the web by integrating Python tools within an R process. We will explain text modeling using R and Python in three parts.

The first part of the tutorial will discuss how to setup a Python installation within an R session as well as how R determines the Python environment to use and how to enforce the right environment for a given problem. Often, you attempt to setup a particular Python environment for a text analytics problem but R finds the wrong version of Python, and your code does not work. We will also discuss how to pass objects back and forth between R and Python as well as how to load and install Python modules from R.

The second part of the tutorial will discuss the various natural language processing algorithms in Python, how to choose the appropriate modeling steps, and how to structure your data so that the algorithm works correctly. Much of this part will involve the Python package tensorflow (Abadi et al., 2015) but various Python engines could also apply given problem type and library familiarity. One can think of this section as a crash course in Neural Networks for NLP in Python.

Finally, we will show how to create a natural language processing workflow by reading the data into R, performing basic data manipulations, passing the objects into the appropriate Python environment, running the text analysis algorithm, and displaying/saving the final results from R. To better facilitate learning and applicability to I/O research, we will walk through an example demonstrating modeling the sentiment of an organization in different business units across people of different demographic attributes. Although our example applies to sentiment modeling, any systematic analysis of text data can map onto a similar process.

Audience members are strongly encouraged to bring laptops and to have downloaded the materials ahead of time. For those who wish to follow along, we will make available all materials and R scripts at <https://github.com/swnydick/siop-2022-nlp-r-python>. We request 80 minutes for this tutorial, with the approximate time for each topic as well as additional information provided below. Note that none of the authors are affiliated with the producers of any of the packages described and that there are no material gains (financial or otherwise) for them. All packages are free-of-charge and complete.

**Topic #1: Integrating Python and R (15-20 minutes)**

Python and R are often thought of as conflicting ecosystems rather than complementary tools. Recently, the R package reticulate (Ushey, Allaire, & Tang, 2021) was developed to facilitate integrating Python within an R workflow. However, using Python within R is still tricky. Python has several package management systems (e.g., pip and Anaconda, Anaconda Software Distribution, 2020), as well as different methods of setting and using environments (e.g., venv and Anaconda, Anaconda Software Distribution, 2020). R users often have difficulty with Pythons package and environment management systems. Python modules often have a complex dependency structure where one module depends on an early version of a dependent module and a later version of a different module. This contrasts with R’s straightforward dependencies, where most packages will work as long as the latest versions of the dependent packages are installed. Therefore, one must understand, use, maintain, and access environments in Python to ensure that code is transferable (although, see Ushey, 2021, for a package dependency management system in R).

Even though one can access R objects in Python, objects mean different things in each languages. The reticulate package converts the most common objects across languages (e.g., vector in R results in list in Python, named list in R results in dict in Python), but even though most obvious conversions result in objects with different properties. For instance, an atomic vector in R has objects all of the same type, but a list in Python can be of various types. Moreover, Python lists are assigned by reference to the object, so that modifying one instance of a list will modify any other instance. Understanding how Python collections work (of which lists are a primary example) can be difficult for R users who assume that assigning variable “a” to variable “b” copies the object rather than merely pointing to the same memory location (see R environments for a counter example; most R users do not modify environments directly, but they play apart in the magic of shiny applications, see Wickham, 2021; R environments as a language property are different from the aforementioned Python environments as a system for organizing dependencies, although the latter does relate to the renv R package, see Ushey, 2021).

The first portion of the tutorial will be a crash course in how to setup and access Python environments as well as a brief overview of how R and Python differ in how they generally work.

**Topic #2: Neural Network Models with R and Python (30-35 minutes)**

Text data can lead to various questions, such as: how do people in various groups feel about this organization or these work processes; or what is the general topic of conversation on this pre-employment hiring board? A sophisticated way of finding patterns in complex data is trying to replicate the reasoning process that makes “is this a cat” much easier for a person than a computer. After cleaning the initial data (see Jones, Nydick, & Wiseman, 2021a for a walkthrough of initial text processing steps), text can be coded as embeddings that treats semantically similar words as numerically similar (where similar means vectors that are close to each other in space; see Mikolov, Chen, Corrado, & Dean, 2013 for description of the word2vec method). One could train embedding models based on the similarity among words in many text corpuses. However, this process is incredibly computationally intensive; moreover, many sophisticated and open source embedding models exist, including word2vec, GloVe, and BERT, so that practitioners need not create embedding matrices from scratch.

After cleaning data and coding the text as useful numbers, these numbers can then be fed into a neural network with a set of layers based on the proposed structure of how the words should be processed when predicting some response, including dense feed-forward layers (all neurons in current layer impact all neurons in next layer), convolution layers (reducing an object to simpler building blocks), long short-term memory layers (to efficiently retain complex dependencies among words in different locations), and many more (see Tch, 2017, for a more complete and detailed explanation). These networks can be linked together to predict an outcome, such as sentiment or sentence topic. Many of the commonly-used algorithms for fitting a neural network model (such as Tensorflow) were written for a Python interface.

During this section of the tutorial, we will explain the neural network words and phrases that you need to know, how to apply a neural network structure/algorithm to a given text classification problem, and how to build a basic neural network using the Keras/Tensorflow interface. We will also explain how to build a neural network in R by using the keras (Allaire & Chollet, 2021) package. Note that the keras package uses R code but depends upon having a working version of Python installed with the necessary dependencies.

**Topic #3: Natural Language Processing Workflow (20 minutes)**

During the last major section, we will present a typical workflow for processing natural language data using a question common to I/O psychology: how do people feel about their work environment? We will walk through each of the steps of reading the data into R, performing text cleaning, building a neural network model, making sure that the correct dependencies are installed and available in Python, fitting the model using the R frontend to keras (while knowing how to debug issues in Python), and then clustering/displaying the final result to tell a story on the types of employees expressing negative feelings about their work and actions that could be taken to improve the work environment.

**Topic #4: Wrap-up (5-10 minutes)**

Finally, the presenters will answer audience questions and help with technical problems encountered during previous sections. The presenters will also provide materials for participants to read for self-study and include links to useful materials for solving text analytics problems.

**Learning Objectives**

This workshop is designed to help you:

1. Install, update, and use Python within an R session for the purposes of analyzing and processing text data.
2. Understand how to run state-of-the art machine learning and neural network algorithms in python to analyze unstructured text.
3. Create a workflow so that data can be cleaned and processed in R, analyzed in Python, and then summarized and reported on in R.

**Presenter Information**

Steven Nydick

Senior Manager, Measurement, Data, and Automation

Korn Ferry

33 South Sixth Street

Suite 4900

Minneapolis, MN 55402

Phone: 612-373-3548

Email: [steven.nydick@kornferry.com](mailto:steven.nydick@kornferry.com)

Membership Status: Member

Benjamin Wiseman

Senior Manager, Data Science and AI

Korn Ferry

Email: benjamin.wiseman@kornferry.com

**Presenter Bios**

Steven Nydick is a Senior Manager of Assessment Design and Data Automation at the Korn Ferry Institute, where he designs R-based tools and scoring algorithms. He is the lead author and maintainer of the catIrt R package as well as several internal R packages helping with everything from plotting to powerpoint generation to interfacing with servers. He has contributed to developing psychometric models and corresponding estimation algorithms that have been published in *Applied Psychological Methods* and the *Journal of Educational and Behavioral Statistics*. Steven received his Ph.D at the University of Minnesota in Psychometrics and Quantitative Psychology, where he primarily studied IRT-based adaptive tests for selection and classification. He also has an M.S. in Statistics from the University of Minnesota.

Ben Wiseman is a Senior Data Scientist at the Korn Ferry Institute responsible for maintaining and developing R-based automation tools, models, reports, and user interfaces. He has publications in entomology, ecology, and molecular evolution and has worked with and trained numerous clients in the military, public, and private sectors on a wide range of applications. Ben received his MSc from Lincoln University (New Zealand) in applied statistical modelling where he developed a user-facing geospatial AI platform for DOCs predator monitoring and control systems.

**References**

Abadi, M. et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org.

Allaire, J. J., & Chollet, F. (2021). Keras: R interface to “Keras”. R package version 2.4.0.

Anaconda Software Distribution (2020). *Anaconda Documentation*. Anaconda Inc. Retrieved from <https://docs.anaconda.com>.

Cass, S. (2018). The 2018 top programming languages. Retrieved August 29, 2018, from <https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages>

Cass, S. (2021). The 2021 top programming languages. Retrieved September 7, 2021, from <https://spectrum.ieee.org/top-programming-languages/>

Chollet, F. et al. (2015). *Keras*. GitHub. Retrieved from <https://github.com/fchollet/keras>.

Dowle, M., & Srinivasan, A. (2021). data.table: Extension of `data.frame`. R package version 1.14.0.

Harris, C. R., Millman, K. J., van der Walt, S. J. et al. (2020). Array programming with NumPy. *Nature*, *585*, 357-362.

Hornik, K. (2017). The R FAQ. Retrieved from <http://CRAN.R-project.org/doc/FAQ/R-FAQ.html>

Jones, J. A., Nydick, S. W., & Wiseman, B. (2019a, April). Effective data wrangling and visualization with R. Master Tutorial at the annual meeting of the Society of Industrial and Organizational Psychology, National Harbor, MD.

Jones, J. A., Nydick, S. W., & Wiseman, B. (2019b, April). Web scraping with R. Master Tutorial at the annual meeting of the Society of the Industrial and Organizational Psychology, National Harbor, MD.

Jones, J. A., Nydick, S. W., & Wiseman, B. (2021a, April). Big Data Systems with R. Master Tutorial at

the annual meeting of the Society of the Industrial and Organizational Psychology, New Orleans, LA (Virtual Meeting).

Jones, J. A., Nydick, S. W., & Wiseman, B. (2021b, April). Text Analytics and NLP with R.

Master Tutorial at the annual meeting of the Society of Industrial and Organizational Psychology, New Orleans, LA (Virtual Meeting).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in

Vector spaces. In *Proceedings of the Workshop at ICLR*. Retrieved from <https://arxiv.org/pdf/1301.3781.pdf>

Mourri, Y. B., Kaiser, L., & Shyu, E. (n.d.). *Natural Language Processing Specialization* [MOOC]. Coursera. <https://www.coursera.org/specializations/natural-language-processing>.

Munzert, S., Rubba, C., Meißner, P., & Nyhuis, D. (2015). *Automated Data Collection with R*. New York, NY: Wiley.

The pandas development team. (2021). *Pandas-dev/pandas: Pandas*, *version 1.3.2.* Available at <https://pandas.pydata.org/>.

Paszke, A. et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32* (pp. 8024-8035). Retrieved from <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.

Pedregosa et al. (2011). Scikit-learn: Machine learning in Python. *JMLR*, *12*, 2825-2830.

Python Software Foundation. (2021). *Python Language Reference, version 3.7*. Available at <http://www.python.org>.

R Core Team. (2019). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. URL <https://www.R-project.org/>.

Robinson, D. (2017). The impressive growth of R. Retrieved August 29, 2018, from

<https://stackoverflow.blog/2017/10/10/impressive-growth-r/>

Tch, A. (2017). The mostly complete chart of Neural Networks, explained. Retrieved September 13,

2021, from <https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464>

Ushey, K. (2021). renv: Project Environments. R package version 0.14.0.

Ushey, K., Allaire, J., & Tang, Y. (2021). reticulate: Interface to “Python”. R package version 1.20.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. New York, NY: Springer.

Wickham, H. (2021). Masterhing Shiny. Sebastopol, CA: O’Reilly Media, Inc.

Wickham, H., François, R., Henry, L., & Müller, K. (2021). dplyr: A grammar of data manipulation. R

package version 1.0.6.

**Appendix**

CV Steven Nydick

CV Benjamin Wiseman

Steven Nydick

Email: Steven.Nydick@KornFerry.com

**Education:**

PhD, University of Minnesota, Psychometrics/Quantitative Psychology, 2013.

Advisor: Niels Waller

MA, University of Minnesota, Psychometrics/Quantitative Psychology, 2012.

Advisor: Niels Waller

MS, University of Minnesota, Statistics, 2011.

Advisor: Sanford Weisberg

BS, Syracuse University, Mathematics and Psychology, 2006.

**Professional Experience:**

Senior Manager, Measurement, Data, and Automation, 2021 - Present

Data Scientist Developer, Korn Ferry, 2018 – 2021.

Senior Psychometrician, Pearson VUE, 2016 – 2018.

Psychometrician, Pearson VUE, 2013 – 2016.

Research Assistant, University of Minnesota, 2013 –2019.

Intern in Psychometrics, ARRT, 2012 – 2013.

Intern in Psychometrics, ACT, 2011.

Graduate Instructor/Section Leader, University of Minnesota, 2007 – 2013.

**Awards:**

Doctoral Dissertation Fellowship, 2013

Graduate Research Partnership Program, 2010

Archimedes Prize in Mathematics, 2006

**Manuscripts Published and In Press:**

Wang, C. & Nydick, S. W. (2020). On longitudinal item response theory models: A didactic. *Journal of*

*Educational and Behavioral Statistics*, *45*, 339-368.

Wang, C. & Nydick, S. W. (2015). Comparing two algorithms for calibrating the restricted non-

compensatory multidimensional IRT model. *Applied Psychological Measurement*, *39*, 119-134.

Nydick, S. W. (2014). The sequential probability ratio test and binary item response models. *Journal of*

*Educational and Behavioral Statistics*, *39*, 203-230.

**Software:**

Wiseman, B., Nydick, S. W., & Jones, J. A. (2018). roperators: Additional operators to

help you write cleaner R code. R package version 1.0.1.

<https://CRAN.R-project.org/package=roperators>

Nydick, S. W. (2014). catIrt: An R package for simulating computerized adaptive tests. R package version

0.5-0).

**Presentations and Workshops:**

Jones, J. A., Nydick, S. W., & Wiseman, B. (2021, April). Big Data Systems with R. Master Tutorial at the

annual meeting of the Society of the Industrial and Organizational Psychology, New Orleans, LA (Virtual Meeting).

Jones, J. A., Nydick, S. W., & Wiseman, B. (2021, April). Text Analytics and NLP with R.

Master Tutorial at the annual meeting of the Society of Industrial and Organizational Psychology, New Orleans, LA (Virtual Meeting).

Jones, J. A., Nydick, S. W., & Wiseman, B. (2019, April). Web scraping with R. Master Tutorial at the

annual meeting of the Society of the Industrial and Organizational Psychology, National Harbor, MD.

Jones, J. A., Nydick, S. W., & Wiseman, B. (2019, April). Effective data wrangling and visualization with R.

Master Tutorial at the annual meeting of the Society of Industrial and Organizational Psychology, National Harbor, MD.

Nydick, S. W. (2016, April). The expected likelihood in computerized classification testing. Paper

presented at the annual meeting of the National Council on Measurement in Education, Washington, DC.

Nydick, S. W. (2014, April). Multidimensional mastery testing with CAT. Paper presented at the annual

meeting of the National Council on Measurement in Education, Philadelphia, PA.

Nydick, S. W., Wang, C., & Xiong, X. (2014, April). Measuring multidimensional growth—a higher-order

IRT perspective. Paper presented at the annual meeting of the American Educational Research Association, Philadelphia, PA.

Nydick, S. W., Nozawa, Y., & Zhu, R. (2012, April). Accuracy and efficiency in classifying examinees using

computerized adaptive tests: An application to a large scale test. Paper presented at the Annual Meeting of the National Council on Measurement in Education, Vancouver, BC.

Nydick, S. W., & Weiss, D. J. (2010, June). Accepting the null: No change in change CAT. Paper presented

at the IACAT conference on CAT, Arnhem, NL.

Nydick, S. W., & Weiss, D. J. (2009). A hybrid simulation procedure, evaluated for the development of

CATs. In D. J. Weiss (Ed.) *Proceedings of the 2009 GMAC Conference on Computerized Adaptive Testing.*

**Unpublished Manuscripts:**

Nydick, S. W. (2013). *Intro to R for Psychologists.* Minneapolis, MN: Author.

**Courses Taught:**

Introduction to Data Analysis/Statistics for Undergraduates

Honors Introduction to Data Analysis/Statistics for Undergraduates

Analysis of Psychological Data for Graduate Students

Benjamin Wiseman

Email: Benjamin.Wiseman@KornFerry.com

**Education:**

MS, Lincoln University, Applied Statistics, 2015.

BS, Lincoln University, Biostatistics, 2013.

**Professional Experience:**

Senior Manager, Data Science and AI, Korn Ferry, 2021 – Present.

Data Scientist Developer, Korn Ferry, 2018 – 2021.

Owner, Wiseman Analytics, 2016 – 2018.

Information Services, DHS, 2015 – 2016.

Instructor, Lincoln University, 2013 – 2014.

Research Assistant, Lincoln University, 2011 – 2015.

Research Assistant, Seoul National University, 2011.

**Awards:**

Freemasons university scholarship

Forest and Bird research award

AGLS research scholarship

**Manuscripts Published and In Press:**

Wiseman, BH., Fountain, ED., Bowie, MH. He, S., Cruickshank, RH. 2016. Vivid molecular divergence over volcanic remnants: the phylogeography of Megadromus guerinii on Banks Peninsula, New Zealand. New Zealand Journal of Zoology

Fountain, ED., Pugh, AR., Wiseman, BH., Smith, VR., Cruickshank, RH., and Paterson, AM. 2015. On the captive rearing of Hadramphus tuberculatus (Pascoe 1877) (Coleoptera: Curculionidae: Molytinae):is ex-situ conservation the lesser of two weevils? New Zealand Entomologist.

Gillespie, M., Cruickshank, RH., Wiseman, BH., Wratten, S. 2013. Incongruence between morphological and molecular markers in the butterfly genus Zizina (Lepidoptera: Lycaenidae) in New Zealand.Systematic Entomology 38:151-163.

Fountain, ED., Wiseman, BH., Cruickshank, RH., and Paterson, AM. 2013. The ecology and conservation of Hadramphus tuberculatus (Pascoe 1877) (Coleoptera: Curculionidae: Molytinae). Journal of Insect Conservation 17:737-745.

**Software:**

Wiseman, B. W., Nydick, S.W., Jones, J (2018) roperators: Additional Operators to Help you Write Cleaner R Code. R package version 1.0-1).

Wiseman, B. W. (2015) Neurofriendly: Artificial Neural Networks Made Simple

Wiseman, B. W. (2015) Geofriendly: Easy Spatial Application of Artificial Neural Networks

**Presentations and Workshops:**

Jones, J. A., Nydick, S. W., & Wiseman, B. (2021, April). Big Data Systems with R. Master Tutorial at the

annual meeting of the Society of the Industrial and Organizational Psychology, New Orleans, LA (Virtual Meeting).

Jones, J. A., Nydick, S. W., & Wiseman, B. (2021, April). Text Analytics and NLP with R.

Master Tutorial at the annual meeting of the Society of Industrial and Organizational Psychology, New Orleans, LA (Virtual Meeting).

Jones, J. A., Nydick, S. W., & Wiseman, B. (2019, April). Web scraping with R. Master Tutorial at the

annual meeting of the Society of the Industrial and Organizational Psychology, National Harbor, MD.

Jones, J. A., Nydick, S. W., & Wiseman, B. (2019, April). Effective data wrangling and visualization with R.

Master Tutorial at the annual meeting of the Society of Industrial and Organizational Psychology, National Harbor, MD.

Wiseman, B. H. 2017 Data Science with Python. ESRI Developer Summit, Palm Springs, CA.

Wiseman, B. H. 2013 Messy data, messy models and applied statistics. Presented for Bio-Protection seminar, Lincoln University, New Zealand.

Marris, J. and Wiseman, B. H. 2012. Islands in the snow: Ecology, systematics and biogeography of the New Zealand beetle genus Protodendrophagus (Coleoptera:Silvanidae:Brotini). Presented at the New Zealand Ecological Society conference.

Cripps, M., McNeil, M., Patrick, H., Wiseman, B., Nobilly, F., Edwards, G. 2012. Invertebrate abundance and diversity in intensively managed dairy pastures. New Zealand Plant Protection Society Conference.

Wiseman, B. H., Cruickshank, R. H., Bowie, M. H., Fountain, E. D. 2011. Unexpected genetic variation in an endemic ground beetle: The molecular mystery of Megadromus guerinii (Coleoptera: Carabidae). 3rdAnnual Combined Australian and New Zealand Entomological Societies Conference

Wiseman, B. H. (2011). The curious case of Megadromus guerinii: phylogeographic oddities on Bank’s Peninsula. Presented to the Canterbury branch of the New Zealand Entomological Society.

**Courses Taught:**

Research and Analytical Skills

Geospatial Information Systems with Arc GIS

Business Statistics

Intermediate Statistics for Commerce