ASMS 2018
ANNUAL
CONFERENCE
WORKSHOP

SO.CAL

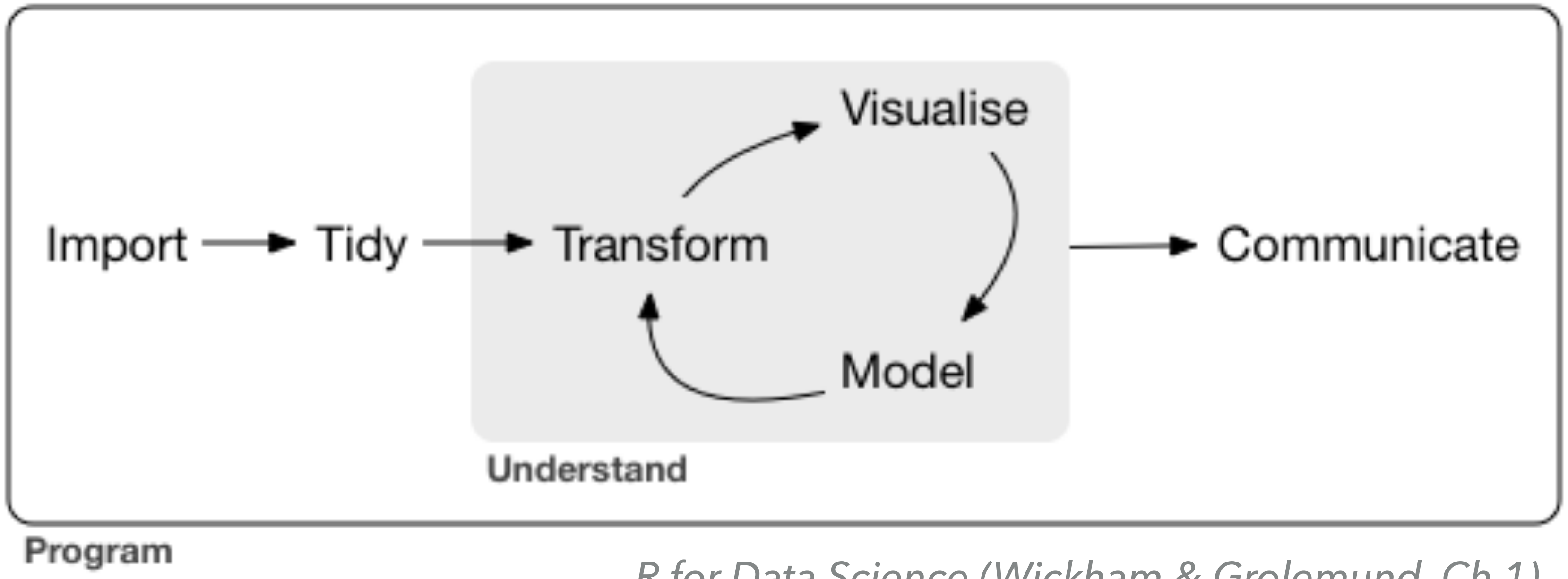
SOCALBIOINFORMATICS.COM

USING R FOR MASS SPECTROMETRY DATA ANALYSIS & WORKFLOWS

WORKSHOP OVERVIEW

- Main Goal

  present and discuss tools within the R ecosystem that support mass spec data analysis & analysis workflows



R for Data Science (Wickham & Grolemund, Ch 1)

WORKSHOP OVERVIEW

- Outline
  - Introductory tips, thoughts & suggestions
  - Converting MS data files to open formats
  - R packages for MS analysis
  - tidyverse for analysis workflows
  - Shiny apps for data exploration and communication
  - Bringing everything together

WORKSHOP OVERVIEW

All Presentation Material Are on GitHub

https://github.com/ZenBrayn/asms-2018-r-workshop

GETTING STARTED

WIN/MAC/LINUX: INSTALL R & RSTUDIO

- Install R first!
  https://cloud.r-project.org

- Then install RStudio Desktop
  https://www.rstudio.com/products/rstudio/download/

- Install R packages (easy to do through RStudio)
  Tools → Install Packages…

NOTES ON INSTALLING PACKAGES

- Some packages require shared libraries and/or compilation
- On the Mac: Install Xcode (from the Mac App Store) and the Command Line Tools
(from the terminal run: `xcode-select --install`)
- Pay attention to any error messages
They're getting better and could be helpful!
- Google is your friend!

EXAMPLE PACKAGE ERROR MESSAGE

```
...
Using PKG_CFLAGS=
Using PKG_LIBS=-lxml2
------------------------- ANTICONF ERROR ---------------------------
Configuration failed because libxml-2.0 was not found. Try installing:
 * deb: libxml2-dev (Debian, Ubuntu, etc)
 * rpm: libxml2-devel (Fedora, CentOS, RHEL)
 * csw: libxml2_dev (Solaris)
If libxml-2.0 is already installed, check that 'pkg-config' is in your
PATH and PKG_CONFIG_PATH contains a libxml-2.0.pc file. If pkg-config
is unavailable you can set INCLUDE_DIR and LIB_DIR manually via:
R CMD INSTALL --configure-vars='INCLUDE_DIR=... LIB_DIR=...'
--------------------------------------------------------------------
ERROR: configuration failed for package 'xml2'
* removing '/usr/local/lib/R/site-library/xml2'
ERROR: dependency 'xml2' is not available for package 'tm'
* removing '/usr/local/lib/R/site-library/tm'

The downloaded source packages are in
     '/tmp/RtmpLb48pu/downloaded_packages'
Warning messages:
1: In install.packages("tm") :
  installation of package 'xml2' had non-zero exit status
2: In install.packages("tm") :
  installation of package 'tm' had non-zero exit status
```

This is actually helpful!

WINDOWS: INSTALL THE LINUX SUBSYSTEM

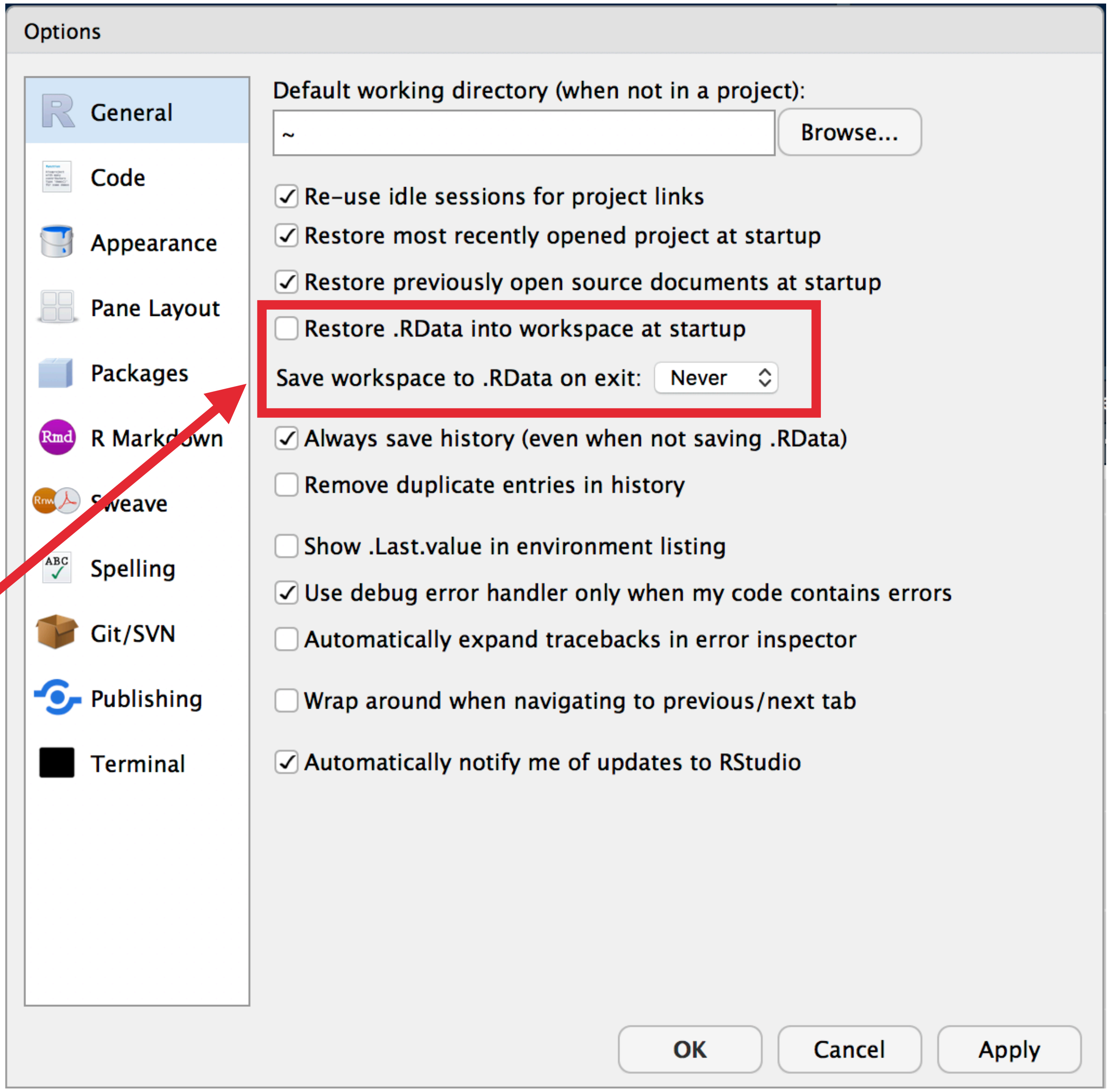
- Follow the guide here

  https://docs.microsoft.com/en-us/windows/wsl/install-win10

- For the current version of Windows 10
  - Enable the Linux subsystem
    need to run a command in the terminal, with admin privileges
  - Install your Linux distro of choice from the Microsoft Store
  - Run the Linux app, create your account, ready to go!

RSTUDIO TIPS

- Learn the keyboard shortcuts & customize them as needed
- Spend most of your time in the code editor, not the console
- knit'ing rmarkdown and running Shiny apps is just a button click away
- Don't save/restore .RData sessions!



Options

Default working directory (when not in a project): ~ Browse...

☑ Re-use idle sessions for project links
☑ Restore most recently opened project at startup
☑ Restore previously open source documents at startup
□ Restore .RData into workspace at startup
Save workspace to .RData on exit: Never
☑ Always save history (even when not saving .RData)
□ Remove duplicate entries in history
□ Show .Last.value in environment listing
☑ Use debug error handler only when my code contains errors
□ Automatically expand tracebacks in error inspector
□ Wrap around when navigating to previous/next tab
☑ Automatically notify me of updates to RStudio

THOUGHTS ON REPRODUCIBLE DATA ANALYSIS

- Could you repeat an analysis 1 month, 6 months, 1 year from now and *know* that you could get the same results?
- Factors to consider for doing reproducible analysis
  - Encapsulate all analysis steps in scripts
  - Red flags: doing analysis in a GUI, copy-paste data
  - Think about how OS and software versions and updates could affect previous analyses
  - Could your input data change or disappear?

A (VERY) BASIC REPRODUCIBLE ANALYSIS WORKFLOW

1. Create a new directory for your analysis project to hold your analysis scripts and outputs

2. Encapsulate ALL data processing and analysis steps in a set of scripts
   • Prefix script names with numbers indicating the order in which to run them (e.g. 01_process.R, 02_analyze.R)
   • Start with "raw" data, do all data processing, manipulation, clean, reformatting *in code*

3. Add a README file to your project directory to document the when/what/why/how of your analysis; keep it current

4. Reproducible Test: move your directory to another location, or ideally a different computer – *can you re-run your analysis and get the same results?*

CONVERTING VENDOR DATA TO OPEN FORMATS

CONVERTING VENDOR DATA TO OPEN FORMATS

- ProteoWizard is the package of choice!

  http://proteowizard.sourceforge.net

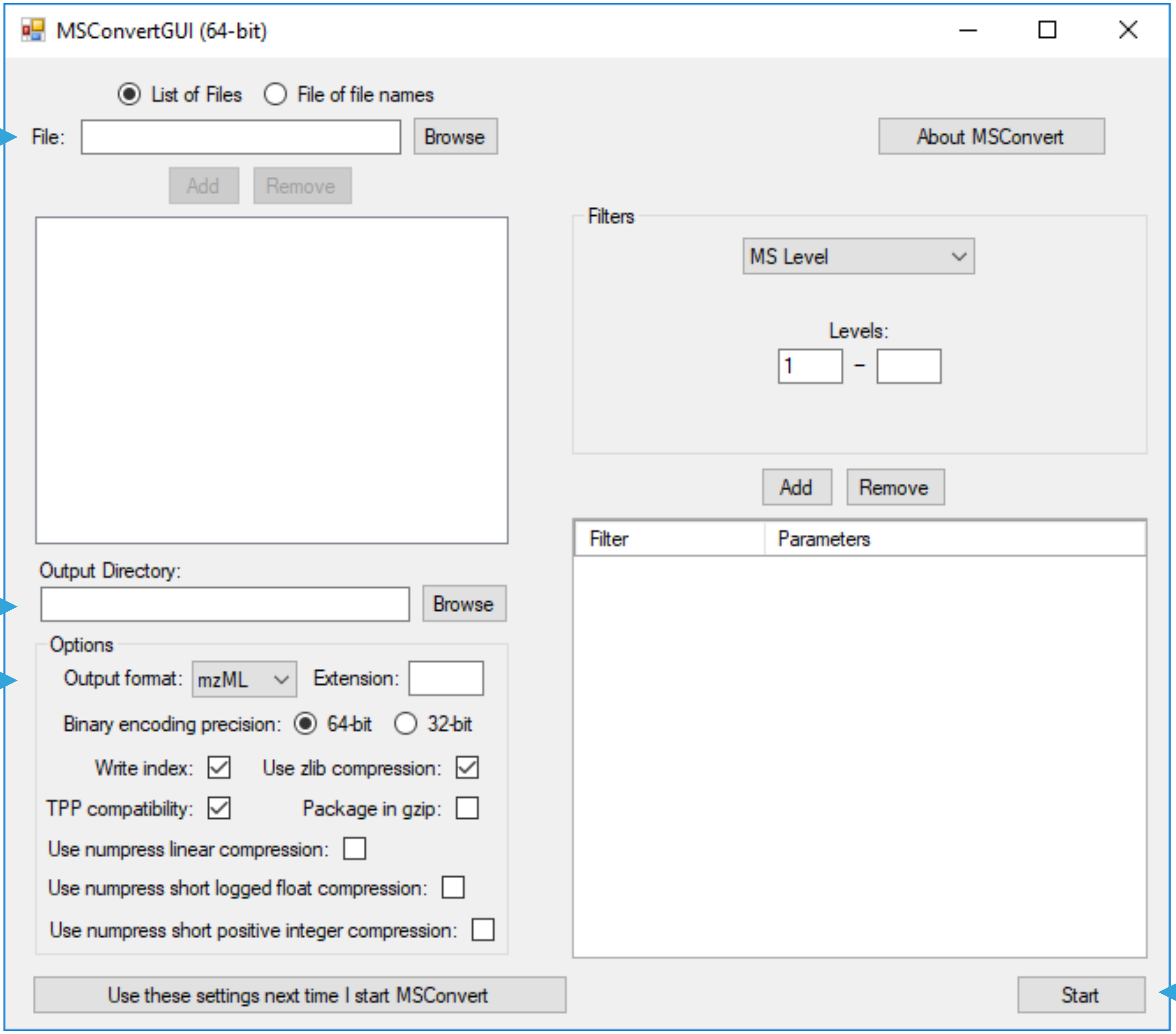
- Provides LOTS of tools for MS data conversion and analysis
- Tool of particular note: **MSConvert**
  a command line and GUI tool for converting among MS data formats including vendor → open formats
- Important: must use the *Windows* version to convert proprietary vendor formats

THE MSCONVERT GUI IS VERY EASY TO USE …

1. Specify input files

2. Specify the output directory

3. Select the output format



MSConvertGUI (64-bit)

● List of Files ○ File of file names

File:

Output Directory:

Options

Output format: mzML Extension:

Binary encoding precision: ● 64-bit ○ 32-bit

Write index: ☑ Use zlib compression: ☑

TPP compatibility: ☑ Package in gzip: □

Use numpress linear compression: □

Use numpress short logged float compression: □

Use numpress short positive integer compression: □

Filters

MS Level

Levels: 1 –

Filter	Parameters

4. Click to convert

… BUT USE THE COMMAND LINE INTERFACE FOR REPRODUCIBLE WORKFLOWS

- MSConvert can also be used through *Command Prompt*

  http://proteowizard.sourceforge.net/tools/msconvert.html

```
# Convert all .RAW files to mzML files and save to output_dir
msconvert *.RAW -o output_dir —mzML

# Might need to specify full path to msconvert, i.e.
“C:\Program Files\ProteoWizard\ProteoWizard 3.0.11252\msconvert.exe”
```

- Also provides many other options for data processing and filtering (see the documentation link above)

YOU CAN INSTALL PROTEOWIZARD IN LINUX ON WINDOWS

- For Ubuntu, first run the Ubuntu app and:

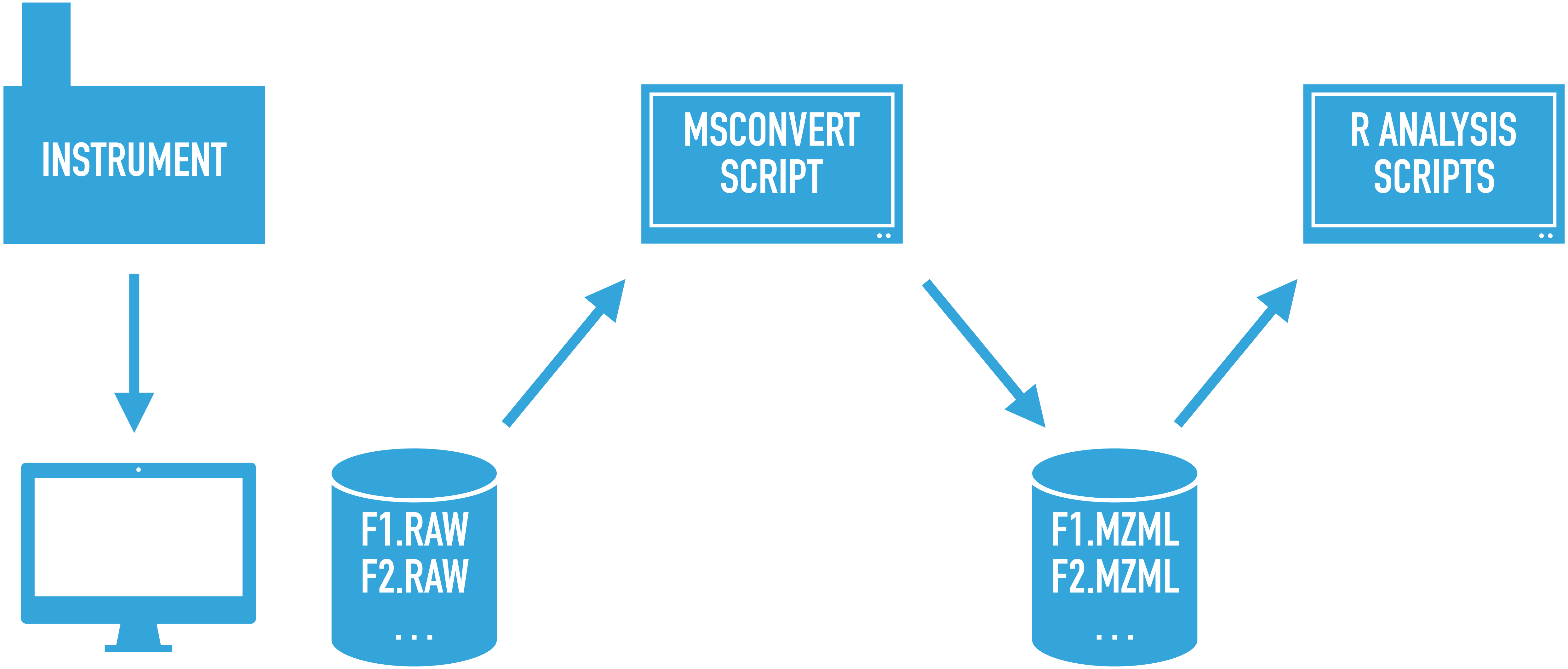
```
# Update apt repositories
sudo apt update

# Install the ProteoWizard tools
sudo apt install libpwiz-tools
```

- IMPORTANT: You can’t convert vendor format files since this is the Linux version
  Use *Command Prompt* to run the Windows version instead

- Otherwise, you can use all the other ProteoWizard tools as usual

EXAMPLE DATA CONVERSION WORKFLOW



OVERVIEW OF R PACKAGES FOR MS DATA PROCESSING AND ANALYSIS

NOTABLE BIOCONDUCTOR R PACKAGES FOR MS DATA ANALYSIS

- **MSnBase**
  infrastructure for reading, processing and analyzing MS data
- **mzR**
  unified API for reading a variety of MS data formats
- **MassSpecWavelet**
  MS spectrum processing tools
- **xcms**
  comprehensive set of tools for MS analysis

Check out the documentation and vignettes on the Bioconductor package pages

XCMS HAS A GREAT TUTORIAL ON LCMS PROCESSING/ANALYSIS

- LCMS data preprocessing and analysis with xcms

  http://bioconductor.org/packages/release/bioc/vignettes/xcms/inst/doc/xcms.html

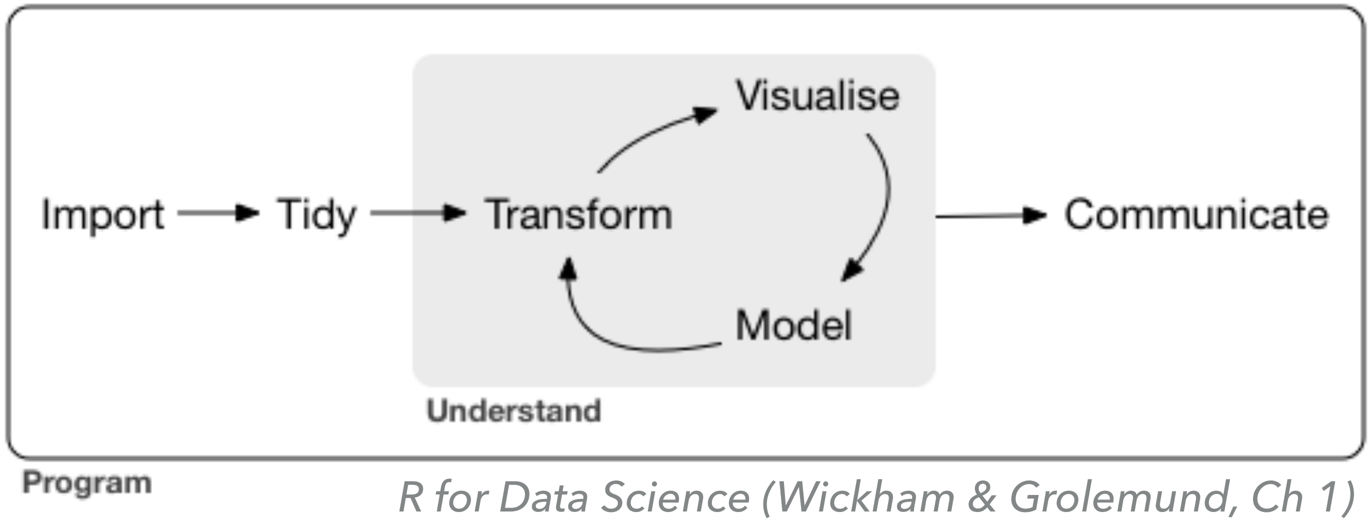
- Covers
  - Loading data ("on disk")
  - High-level data review
  - Chromatographic peak detection
  - Retention time alignment, and cross-experiment feature grouping

USING THE TIDYVERSE FOR ANALYSIS WORKFLOWS

WHAT IS THE TIDYVERSE?

- tidyverse: a collection of R packages designed around the central idea of “tidy data”
  https://www.tidyverse.org
  https://vita.had.co.nz/papers/tidy-data.pdf

- Supports the entire data analysis process in a systematic way, making it easier to write and understand



R for Data Science (Wickham & Grolemund, Ch 1)

WHICH PACKAGES ARE PART OF THE TIDYVERSE?

- Core packages
  - ggplot2: visualization
  - dplyr: data manipulation, data pipelines
  - tidyr: data formatting and arranging
  - readr: data importing from flat files
  - purrr: functional programming toolkit (replace for loops!)
  - tibble: modern data frames
  - stringr: string manipulation
  - forcats: tools for factors (categorical variables)

- LOTS of other associated packages, such as…
  - readxl: data importing from Excel
  - DBI: connecting to databases
  - lubridate: tools for handling dates, times

DPLYR FOR ANALYSIS WORKFLOWS

- dplyr allows you to create powerful data analysis **workflows** based upon data manipulation **verbs**
  - *mutate*: add new variables (columns)
  - *select*: pick specific variables (columns)
  - *filter*: subset to specific cases (rows)
  - *summarize*: transform multiple values to a single summary
  - *arrange*: reorder cases (rows)
  - *group_by*: group data into sub-groups, operate on them individually
- Uses the pipe operator (%>%) to chain expressions together, makes things easier to read and understand

NOTES ON THE TIDYVERSE

- The tidyverse does not replace base R
  - peacefully co-exist, both compliment each other
  - base R functions can usually be used with tidyverse tools

- The tidyverse tools typically use tibbles instead of data frames
  - like data frames, but act more consistently
  - provide powerful additions not possible with data frames
  - some non-tidyverse packages don't like tibbles, you can always convert a tibble to a standard data frame

EXAMPLE: ANALYSIS WITH DPLYR

Live Demo

01_dplyr_example

BUILDING INTERACTIVE DATA APPLICATIONS WITH SHINY

SHINY ALLOWS R USERS TO EASILY BUILD INTERACTIVE DATA APPS

- **Shiny**: an R framework for building interactive data applications, accessed via a web browser
- If you know R, you're 90% of the way towards building (basic) Shiny applications
- Example use cases:
  - Dashboards
  - Data tools/widgets
  - Communicate data to non-analysts, allow them to explore data

THE MAIN STRUCTURE OF A SHINY APP

ui.R – define the application’s interface

```r
library(shiny)

# Define UI for application that draws a histogram
shinyUI(fluidPage(

  # Application title
  titlePanel("Old Faithful Geyser Data"),

  # Sidebar with a slider input for number of bins
  sidebarLayout(
    sidebarPanel(
       sliderInput("bins",
                   "Number of bins:",
                   min = 1,
                   max = 50,
                   value = 30)
    ),

    # Show a plot of the generated distribution
    mainPanel(
       plotOutput("distPlot")
    )
  )
))
```

server.R – define the application’s logic

```r
library(shiny)

# Define server logic required to draw a histogram
shinyServer(function(input, output) {

  output$distPlot <- renderPlot({

    # generate bins based on input$bins from ui.R
    x    <- faithful[, 2]
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    # draw the histogram with the specified number of bins
    hist(x, breaks = bins, col = 'darkgray', border = 'white')

  })

})
```

Also possible to create a single file Shiny application in an app.R file

LEARNING SHINY

- Check out the tutorials

  https://shiny.rstudio.com/tutorial/

- Reactivity is a critical concept for building shiny apps; might be strange at first, but worth learning
- Start simple, don't focus (too much) on the interface, just try to build something useful
- Basic Shiny apps can look bland, but the framework is rich & based on web standards, can do most anything you want

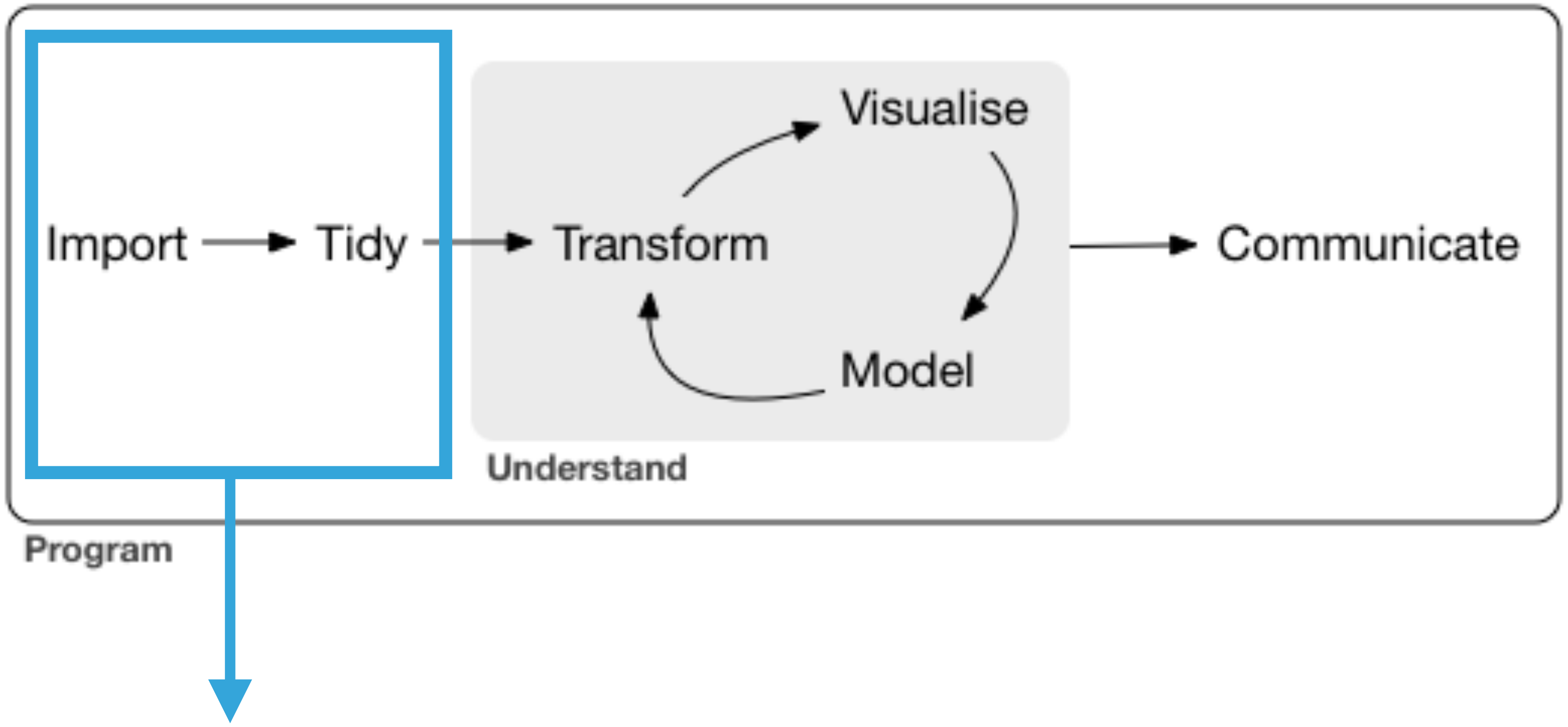
BRING IT ALL TOGETHER: EXAMPLE ANALYSIS WORKFLOW

EXAMPLE SCENARIO

- You've just finished running a study consisting of multiple LCMS runs (different subjects, sample types, etc.)
  - in our example here: 2 subjects, 10 measurements each
- You are tasked with reviewing the data & performing a high-level assessment
- You co-workers in the lab and your supervisor are also interested to see what the data look like

1. READ AND CLEAN YOUR DATA

R for Data Science (Wickham & Grolemund, Ch 1)



- Use ProteoWizard to convert data from raw to open format
- Use mzR to read and parse data files
- Use dplyr and tidyverse to tidy your data

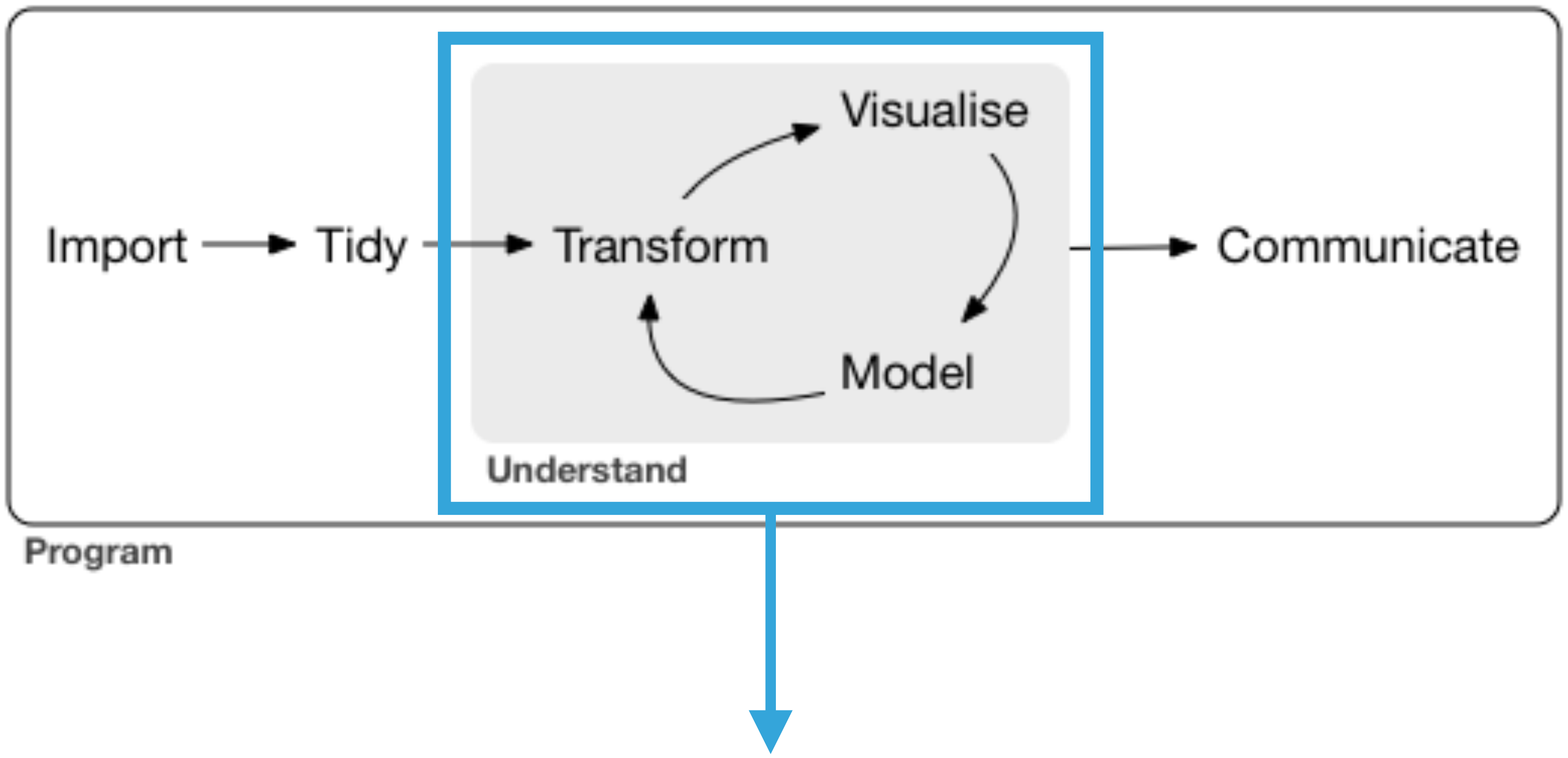
EXAMPLE: PARSING & TIDYING THE LCMS DATA

Live Demo

02_parse_experiments

2. REVIEW AND UNDERSTAND YOUR DATA

R for Data Science (Wickham & Grolemund, Ch 1)



- Use dplyr and tidyverse to explore/drill-in/summarize data
- Use ggplot2 to visualize data (exploratory analysis)
- Look for potential problems, interesting patterns

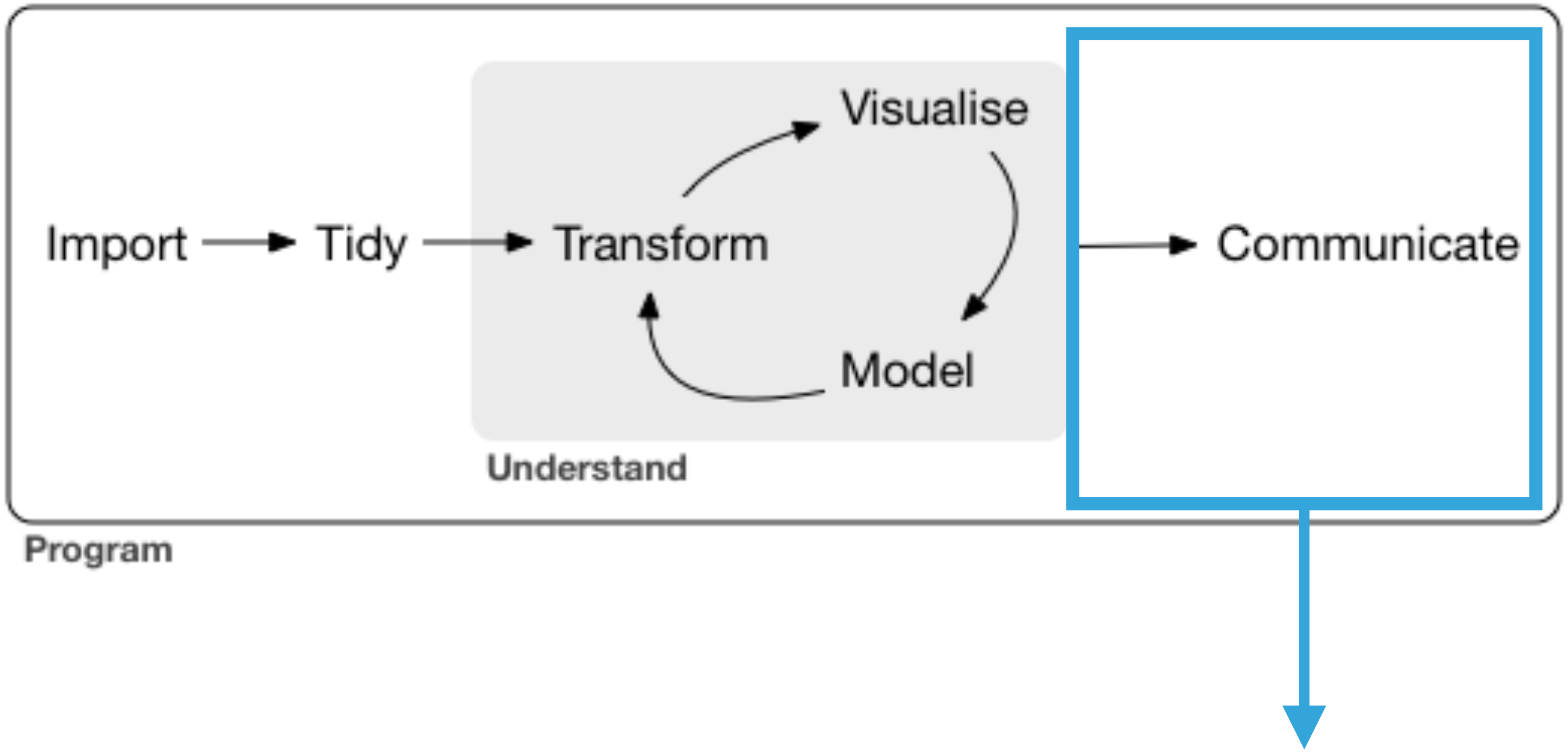
EXAMPLE: REVIEWING THE LCMS DATA

Live Demo

03_review_experiments

3. SHARE YOUR DATA AND RESULTS

R for Data Science (Wickham & Grolemund, Ch 1)



- Use rmarkdown to generate reports
- Use Shiny to create interactive data applications
- Share with others, discuss, listen to feedback

EXAMPLE: SHARING THE LCMS DATA

Live Demo
03_shiny_data_table
04_spectrum_viewer

RESOURCES

- **Books**
  - R for Data Science
  http://r4ds.had.co.nz

  - Applied Predictive Modeling
  http://appliedpredictivemodeling.com

  - ggplot2: Elegant Graphics for Data Analysis, 2nd Ed.

  - Advanced R
  https://adv-r.hadley.nz

- **Websites**
  - RStudio Community
  https://community.rstudio.com

  - Kaggle
  https://www.kaggle.com

  - R Bloggers
  https://www.r-bloggers.com

- **Podcasts**
  - Data Framed
  https://www.datacamp.com/community/podcast