**Preview Questions**

1. What will this book teach me?
2. How is the book organized?
3. What won’t the book teach me?
4. What base level of programming experience must I have to be successful?
5. How do I run R code?
6. Where can I get help with the concepts covered in the book?
7. What is special about the online version?
8. What are the conventions used in the book
9. How should I use the code examples provided in the book?

**Summary Notes**

*Welcome {Web}*

* The book teaches how to do data science using the R programming language.
* Key tasks
  + Import data into R
  + Structure the data properly
  + Transform the data
  + Visualize the data
  + Model the data
* Key concepts
  + Grammar for graphics
  + Literate programming
  + Reproducible research
* Additional key skills
  + Cognitive frameworks to wrangle, visualize, and explore data

*What You Will Learn*

* Develop a solid foundation in the most important tools of data science.
* Workflow model
  1. Import (load data into a data frame in R)
  2. Wrangle
     1. Tidy (store data in a consistent form)
     2. Transform (narrowing in on observations, creating variables, calculating summary statistics)
  3. Visualize (map data for human interpretation; this step does not scale)
  4. Model (develop mathematical or computational tools to answer precise questions)
  5. Iterate through steps 3-5 as necessary
  6. Communicate (inform those who can take action based on the analysis)

*How This Book is Organized*

* Starts with transformation and visualization so that readers will understand the value without being disenchanted by the boredom of importing and tidying data.
* Easier to understand models if you have an understanding of visualization, tidy data, and programming.

*What You Won’t Learn*

* How to work with big data (i.e., data that is larger than 10 GB).
* Big data problems could be small data problems in disguise (i.e., use subset, subsample, multiple small data problems, etc.)
* Python, Julia, or other programming languages that are useful for data science.
* The book recommends mastering one tool at a time.
* R is specifically designed to support data science (i.e., more than just a programming language).
* Data that does not fit within the paradigm of variable and observation (i.e., rectangular data).
* Hypothesis confirmation (i.e., confirmatory analysis).
* Note that hypothesis generation (i.e., data exploration) looks at data more than once and hypothesis confirmation looks at data only one time.
* Models are useful for hypothesis generation as well as hypothesis confirmation.
* Visualizations are useful for hypothesis confirmation as well as hypothesis generation.

*Prerequisites*

* Numerical literacy and some programming experience.
* R-related tools from the comprehensive R archive network (CRAN) including R, RStudio, R packages from the tidyverse, data packages form outside the tidyverse.

*Running R Code*

* > is called the prompt
* Functions are followed by parentheses (e.g., sum(), mean(), etc.)
* Other R objects (i.e., data or function arguments) do not have parenthesis.

*Getting Help and Learning More*

* The troubleshooting process if you get stuck:
  + Perform a Google search and add “R” to the query to narrow the results.
  + Create and reprex (i.e., representative example).
  + Submit query with reprex to Stackoverflow.
* Invest a little time in learning R everyday.
  + Read the RStudio blog
  + Follow @rstudiotips on Twitter
  + Read <http://www.r-bloggers.com> which aggregates over 500 worldwide blogs about R.

*Using Code Examples*

* Generally, code examples from the book may be used in programs and documentation without permission so long as they are minimal in nature.
* Attribution is appreciated but not required.

**Preview Questions**

**Summary Notes**

Web – Ch. 2 Introduction

* Data exploration 🡪generate many promising leads
  + Looking at data
  + Rapidly generating hypotheses
  + Quickly testing generated hypotheses
* Visualization
  + Learn basic structure of a ggplot2
  + Learn techniques for turning data into plots
* Exploratory data analysis combines visualization and transformation to ask and answer interesting questions about the data.
* Model is not addressed in the beginning because it requires additional skills in data wrangling and programming.

Web – Ch. 3 Visualization

* ggplot2 is one of several R systems for making graphs.
* ggplot2 is considered the most elegant and versatile.
* Data frame is a rectangular collection of variables (in columns) and observations (in rows).
* Complete a ggplot2 graph by adding one or more layers.
* Mapping argument is always paired with [aes()].
* Basic template  
  ggplot(data = <DATA>) +  
   <GEOM\_FUNCTION>(mapping = aes(<MAPPINGS>))
* An aesthetic is a visual property of the objects in a plot (e.g., size, shape, and color of points)
* Value refers to data; level refers to an aesthetic property.
* Scaling is assigning a unique level of an aesthetic to each unique value of a variable.
* Do not use size for a discrete variable.
* You can set the aesthetics of a geom manually.
  + Name of color as character string.
  + Size of data point in mm.
  + Shape of data point as a number from chart (R has 25 built in shapes).
* Common coding problems
  + Misplaced characters
  + Unmatched parentheses
  + Incomplete expression
  + Putting + in wrong location
* [ESCAPE] aborts processing the current command
* [?function\_name] call up help about the R function (can also select function name and press F1 in RStudio).
* Facets are subplots of one subset of the data.
* Geom is a geometrical object that a plot uses to represent data.
* Different geom functions use different visual objects to represent data.
* Not every aesthetic works with every geom.
* Use global mappings to minimize duplication in code.
* Mappings placed in a geom function are treated as local for the layer.
* A geom uses a stat (i.e., statistical transformation) to calculate new values for a graph based on the raw values of the dataset.
* [?geom\_name] shows the default stat used by the geom.
* Generally, geoms and stats can be used interchangeably.
* Every geom has a default stat; every stat has a default geom.
* Reasons to state stat explicity:
  + Override default stat
  + Override default mapping
  + Draw attention to the statistical transformation in the code
* Color bar chart using color or fill aesthetic.
* Position adjustment is used to stack bar charts.
  + Position = “identity” places each object exactly where if falls in the context of the graph.
  + Position = “fill” makes each set of stacked bars the same height.
  + Position =”dodge” places overlapping objects next to each other.
* Overplotting is when points on a grit overlap each other making it hard to see where the mass of the data is located.
* Position = “jitter” adds small amount of random noise to spread the points.
* Adding randomness makes graph less accurate at small scale but more revealing at large scale.
* Cartesian coordinate system is the default for ggplot2.
* The grammar of graphics enables you to uniquely describe any plot as a combination of:
  + Dataset
  + Geom
  + Mappings
  + Stat
  + Position adjustment
  + Coordinate system
  + Faceting scheme

Web – Ch. 4 Workflow: basics

* You can use R as a calculator
* Create new objects with < - read as “gets value” (keyboard shortcut is {Alt}+{-})
* Assignment statements have the form: object\_name < - value
* Object names must start with letter and can only have letters, numbers, “\_”, and “.”
* Use descriptive object names.
* Inspect an object in R by typing its name.
* Surrounding assignment statement with parentheses causes assignment and print to screen simultaneously.

Web – Ch. 6 Workflow: scripts

* Script editor provides more room to work than the console.
* Open script editor with File > New File > R script (keyboard shortcut {ctrl}+{shift}+{N})
* Experiment in the console and place functioning code in the script editor.
* When working in script editor {ctrl}+{enter} executes the current R expression in the console.
* In script editor {ctrl}+{shift}+{S} executes the complete script in one step.
* Always start script with the needed packages.

Web – Ch. 8 Workflow: projects

* Must decide:
  + What about analysis is “real” (i.e., lasting record of what happened).
  + Where the analysis will “live”.
* Consider R scripts as real instead of the objects listed in the environment pane.
* It’s easier to recreate the environment with R scripts than vice versa.
* Instruct RStudio not to preserve workspace between sessions.
* Use getwd() to print the working directory.
* Don’t set the working directory from within R.
* Always use Linux/Mac style forward slashes in path and directories because the backslash means something special in R.
* Always use relative paths in scripts and never absolute paths because they hinder sharing.
* Tilda points to the home directory (documents directory in Windows).
* Keep all files associated with a project together.