Introduction and Syllabus

1 - Welcome to Programming Fundamentals Using R

The purpose of this class is to introduce students to the principles of programming using the R and RStudio software packages. R and RStudio are powerful and versatile data analysis packages that are freely available. The R community is extensive and is constantly developing new tools that rival and, in many cases, surpass commercial statistical software and graphing packages. This class is designed for students new to programming. While the class focus is on programming in R and RStudio, the programming skills taught are designed so that students can transfer their skills to other programming platforms like ADMB or C.

Programming Fundamentals Using R is self-paced but students are expected to complete the class within six months. The amount of material in the class is about the same as a three credit, one semester class. There are no regularly scheduled office hours but you can email or call the instructor to ask questions or set up a meeting. Most contacts will receive a response within one business day.

1.1 - Instructor

Charlie Belinsky
- belinsky@msu.edu
- 989-272-2623

About me...
I started my career as a Software Engineer for Motorola in Arizona where I developed software for military radios. From there I became a high school teacher in Port Huron, MI where I taught Computer Science, Web Design, and Physical Science. After that, I worked as an Instructional Designer for the College of Education at MSU where my main project was developing the hybrid graduate program. I currently work at the QFC where my primary job is developing online classes, including this one. In my free time you will often find me with a backpack deep in a forest or, as my picture testifies to, hanging out by a glacier in the Canadian Rockies.

1.2 - Technical Support

You can contact me, Charlie, regarding technical issues.

MSU also offers **24 hour technical support** for students. This support includes hardware, software, and D2L.
- 517-432-6200 or 844-678-6200 (toll free)
- email: ithelp@msu.edu

- Make sure you have your MSU ID or MSU Community ID when communicating with support.
- Remember your *MSU Community ID is the email you log into class with* if you don't have an MSU email.

1.3 - Requirements

Students should be familiar with basic statistical concepts and tests that one would find in an introductory statistics class (like linear regressions and ANOVAs). All material for this course is online -- no textbooks are required.

1.4 - Tech requirements

Hardware: Any Windows 7, 8, 10, Mac, or Linux machine -- preferably with a webcam and microphone for videoconferencing. Any computer built since 2010 can handle all the hardware and software requirements of this class.

Browser: Firefox is recommended. You can use any browser updated within the last couple years to view, navigate, and complete the whole course but some of the advanced navigation features can only be used in Firefox. *Internet Explorer is not supported.* The last update to Internet Explorer was released in 2015 and it has been replaced by Edge, which is supported.

Videoconferencing: We use Zoom for our videoconference meetings. The meeting link will be emailed to you prior to the meeting. It is recommend that you download and test your camera and microphone on Zoom before attending an instructor meeting. The easiest way to to this is to go to the Zoom test page. *Note: Zoom's test page will download Zoom for you.*

To test your hardware in Zoom, open Zoom and click ***Settings*** in the **Account Window** (right side- *Fig 1*) and the **Settings Window** will open (left side- *Fig 1*). Go to the ***Audio*** and ***Video*** tabs to test your microphone, speakers, and webcam. If you are using a MAC, the view is different but the buttons are the same.

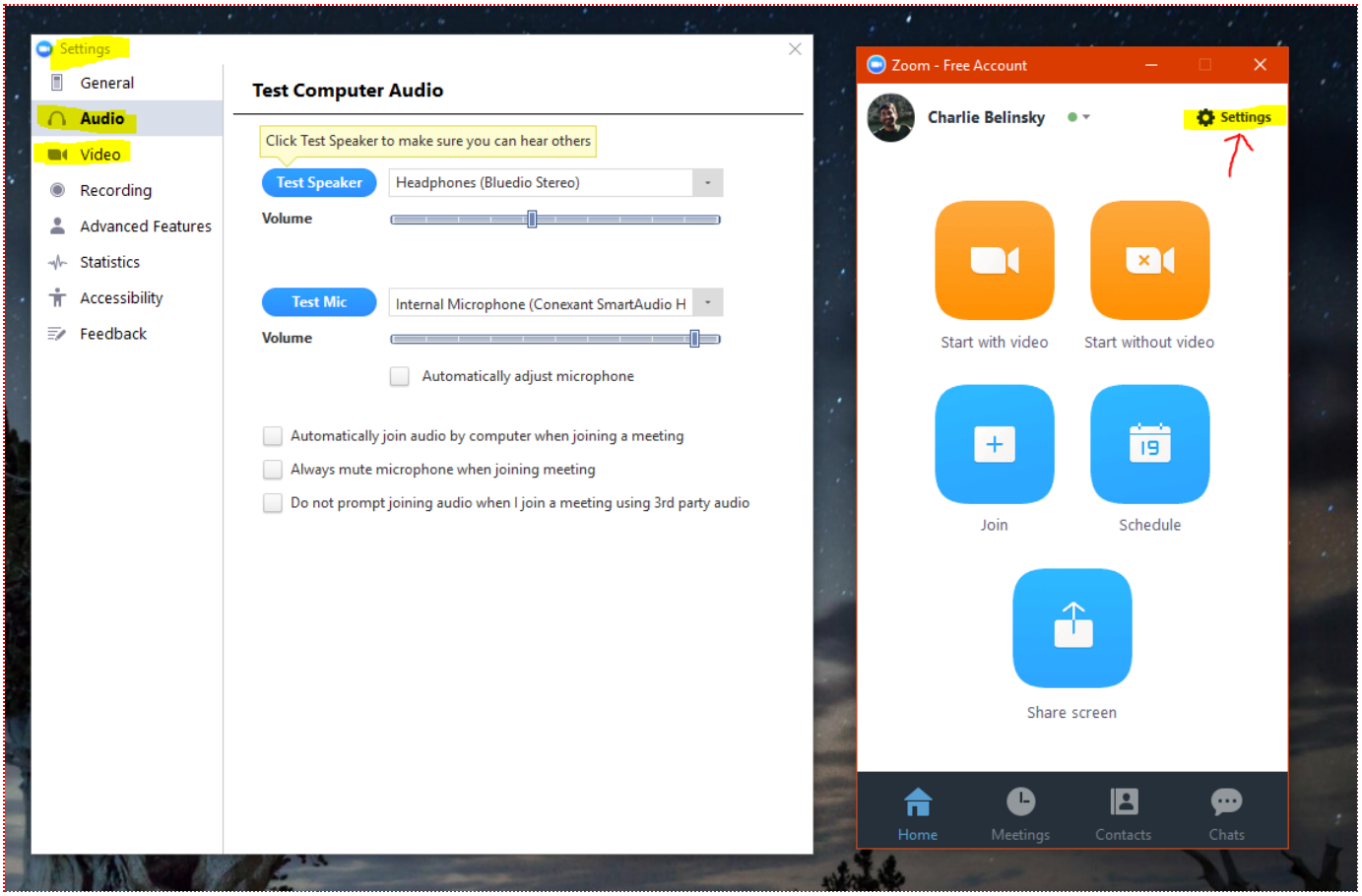

Fig 1: Testing the audio and video in Zoom (click on picture to resize).

MSU ID: You need an MSU ID or an MSU Community ID (guest account) to access the class. You can get an MSU Community ID here. The Community ID is the same as the email you used to get it.

1.5 - Downloading material for the class

All material used in the class is available on GitHub, this includes:

- script and data files used in the lessons -- also linked in relevant lessons
- PDF versions of the lessons that can be used offline
- answers to all the lesson applications, including common programming mistakes

2 - Class Structure

The course has 28 lessons divided into three units: **Basic Programming**, **Data Programming**, and **Statistical Programming**

1. **Basic Programming**: basic programming structures and operations like variables, inputs, outputs, mathematical operations, if-else structures, and conditional operations.
   - **1-1 RStudio Projects:** Set up a Project in RStudio and learn the basic of the RStudio programming environment.
   - **1-2 Variables:** Discussion of the different types of variables, variable naming conventions, assigning values to variables, changing values.
   - **1-3 Mathematical Operations:** Mathematical operations on numeric variables, power operations, algebraic formulas in R
   - **1-4 Debugging:** Using the built-in debugger to find and fix errors in your code.

- **1-5 Strings and Inputs:** Get information from the user and store as a variable, differences between string and numeric variables.
- **1-6 Outputs:** Output information to the console, mixed string and numeric outputs.
- **1-7 Conditional Operators**: Executing code based on a condition, ***if()*** statements, the six conditional operators.
- **1-8 If-Else Statements:** Checking multiple conditions on a variable, creating error conditions.
- **1-9 Multiple Conditions:** Checking complex conditions on a variable using conditional operators && and ||.
- **1-10 Multiple Conditions on Multiple Variables:** More complex conditions, using parentheses to separate conditions.

2. **Data Programming**: data programming structures and operations like file inputs, vectors, subset operations, for loops, functions, and plotting.
   - **2-1 File Inputs:** Creating data files, saving data to a data frame, creating working directories.
   - **2-2 Subset Operations:** Accessing individual data points from a data frame, subset a data frame by name and index.
   - **2-3 Sequences and For Loops:** Iterating through vectors using a sequence and a for loop.
   - **2-4 Iterations and State Variables:** Using Boolean (true/false) variables, mathematical iterations on a vector using a for loop.
   - **2-5 Vector Operations and NA:** Mathematical operations on vectors, summarize basic statistical information from vectors, handling missing data.
   - **2-6 Vectors and Data Frames**: Creating and subsetting vectors, adding vectors to a data frame, manipulating data frames.
   - **2-7 Plotting:** Scatter plots, line plots, styling plots, adding legends.
   - **2-8 Functions:** Creating your own function, adding and using parameters and return values in functions.
   - **2-9 Plotting 2:** Histograms, bar plots, and boxplots.
   - **2-10 Functions 2:** Creating a functions script file, default parameter values, for loops and conditional statements within functions.

3. **Statistical Programming**: reshaping data, sampling, inferential statistics, conditional subsets.
   - **3-1 Packages:** Including packages in your script, getting weather data from NOAA/NCDC database.
   - **3-2 Data Frame Manipulation:** Formatting and reshaping data, getting substrings, mathematical operations on columns of data frames.
   - **3-3 Matrices:** Manipulating vectors, mathematical and statistical operations on reshaped vectors.
   - **3-4 T-tests and ANOVAS:** Comparing means from the weather data using t-tests and ANOVA, boxplots and histograms.
   - **3-5 Sampling**: Random samples, sampling for a normal distribution, creating repeatable pseudo-random values using seed values.
   - **3-6 Linear Regressions:** Scatterplots, scatterplot matrices, linear modeling, fitting linear models.
   - **3-7 Conditional Subsetting:** Subsetting weather data to examine complex relationships.
   - **3-8 Multiple Condition Subsets**: Complex subsetting of data, performing linear regressions on subset data.

Most lessons have an ***Application*** section at the end that will ask you to apply what you learned in the lesson in your own script. The applications are not graded and the answers are provided in the ***Downloads*** section on the home page for the class. However, *it is highly recommended that you email your applications to the instructor for feedback* (emailing instructions are given in the applications).

2.1 - Class Project

The only product that will be evaluated is your class project. Your class project is to use the skills taught in this class with your own data. The class project is meant to be an ongoing assignment that you update as you learn the skills taught in the lessons. If you do not have your own data you can find a dataset online -- there

are many datasets that are freely available online including hundreds at this website. More information and the rubric for the class project is here.

2.2 - Meetings with instructor

Student-instructor meetings will be done on an as-needed basis and meetings can always be requested by the student.

3 - Lesson structure

All lessons have the following structure:

- **Purpose**: Objectives of the lesson.
- **Content**: The material for the lesson.
- **Application**: Practice using the skills taught in the lesson -- answers, and common mistakes, are provided in ***Downloads*** section of home page.
- **Extensions (some lessons)**: Optional content that is related to, but goes beyond, the lesson's objectives.
- **Traps (some lessons)**: Optional content that captures some of the common issues students have with the lesson's objectives.

3.1 - Resizing Pictures

Most pictures in the course can be resized so that the picture is out of your way when you don't need to view it. Clicking on the picture toggles it between the minimized and maximized states (*Fig 2*).

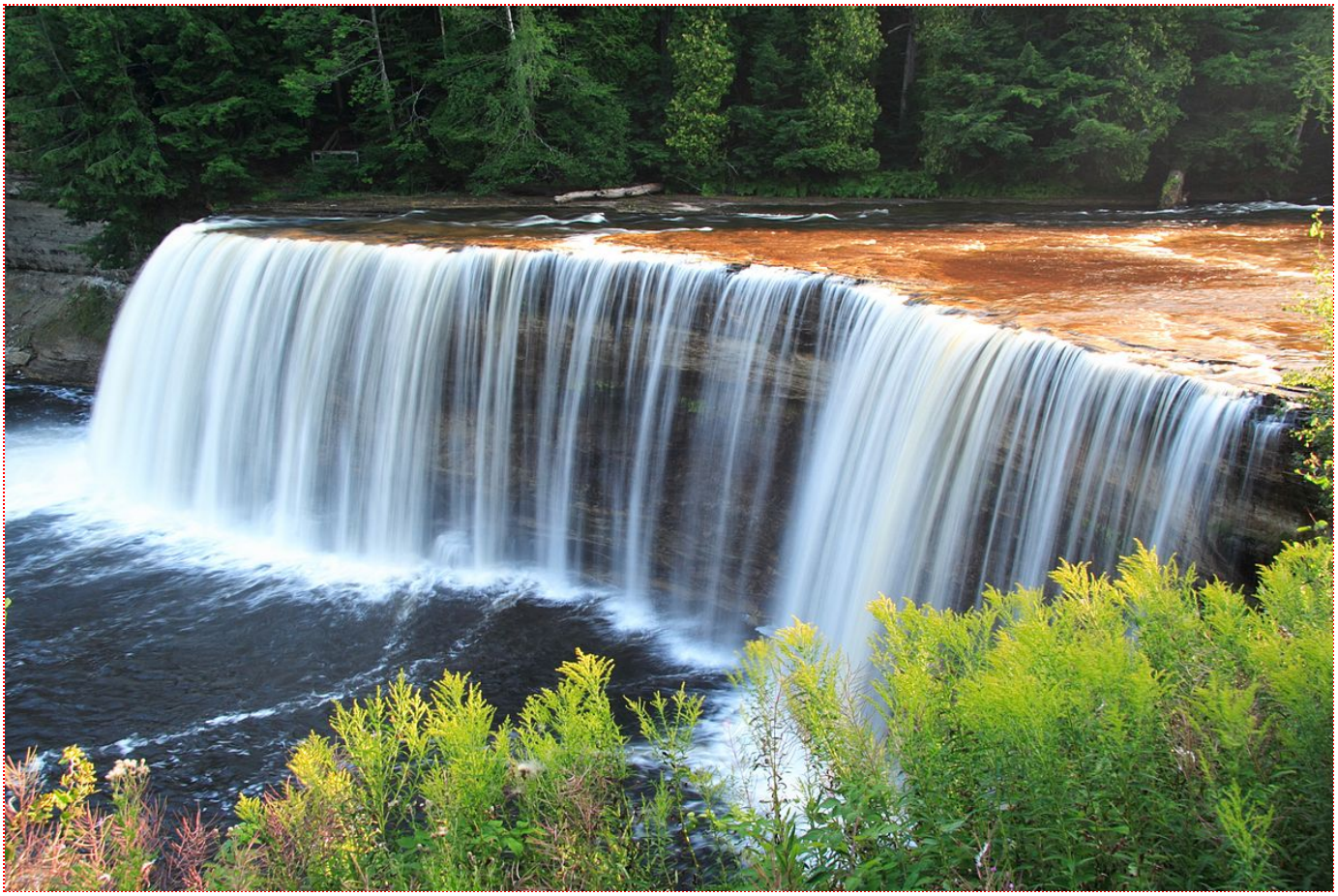

Fig 2: Test picture to toggle (Tahquamenon Falls, Michigan)

3.2 - Extensions and Traps

There are links to ***Extensions*** and ***Traps*** within the Content area of the lesson. Clicking on the appropriate link takes you directly to the ***Extension*** or ***Trap.*** At the end of the ***Extension*** or ***Trap***, another link will appear that takes you back to the original location in the lesson. *Extension: testing the extension link*.

3.3 - Printing lessons and saving to PDF

You can print any lesson or save it as a PDF by clicking on the ***Printer*** *icon* at the top of every lesson (and this syllabus). This will bring up a print dialog (*Fig 3*) and you can print the lesson to a printer. On most machines, you can also choose a PDF device as a printer -- this will save the lesson as a PDF document.

Note: Using your browser's print feature instead on the ***Print*** link will print out the whole webpage instead of just the lesson.

3.3.1 - Print-to-PDF Software

If you are using Windows 7 or 8 you might not have a PDF device and you will need to download Print-to-PDF software. You can see if you have a PDF device by going to the print options in any program and see if any of the devices have "PDF" in their name (*Fig 3*). If you don't have a PDF device, then I recommend you install CutePDF (direct link to the file download). CutePDF is simple print-to-PDF software that does not try to install any extra software on your computer. *Trap: Bloatware*

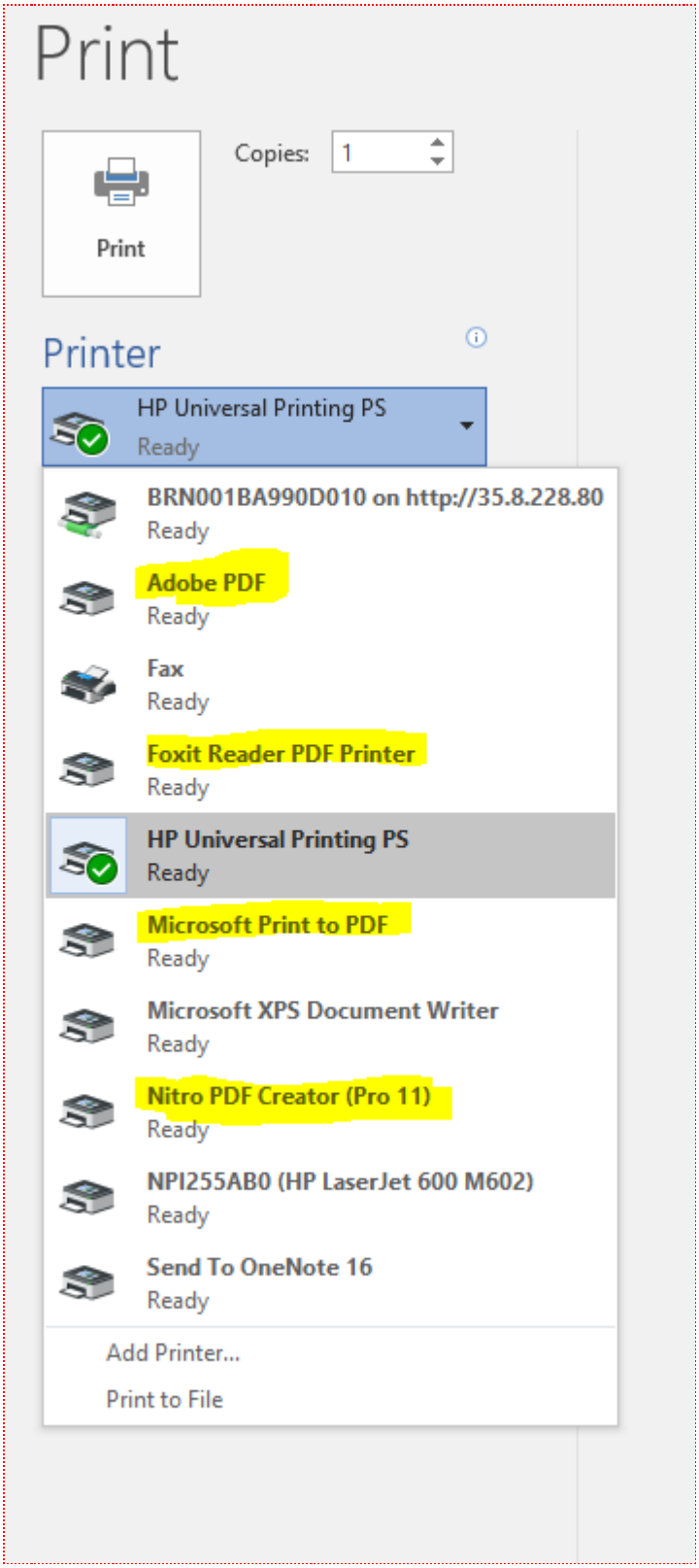

Fig 3: This author has four PDF devices (that is three more than you need!)

3.4 - Right-click features

If you right-click anywhere on the page of any lesson (*Fig 4*), including this one (try it!), you are given options to:

1. print the lesson (same as the lesson's ***Print*** link)
2. maximize all pictures

3. minimize all pictures
4. edit page (for admins only)
5. page map (link to any section of the page -- currently only works using Firefox)

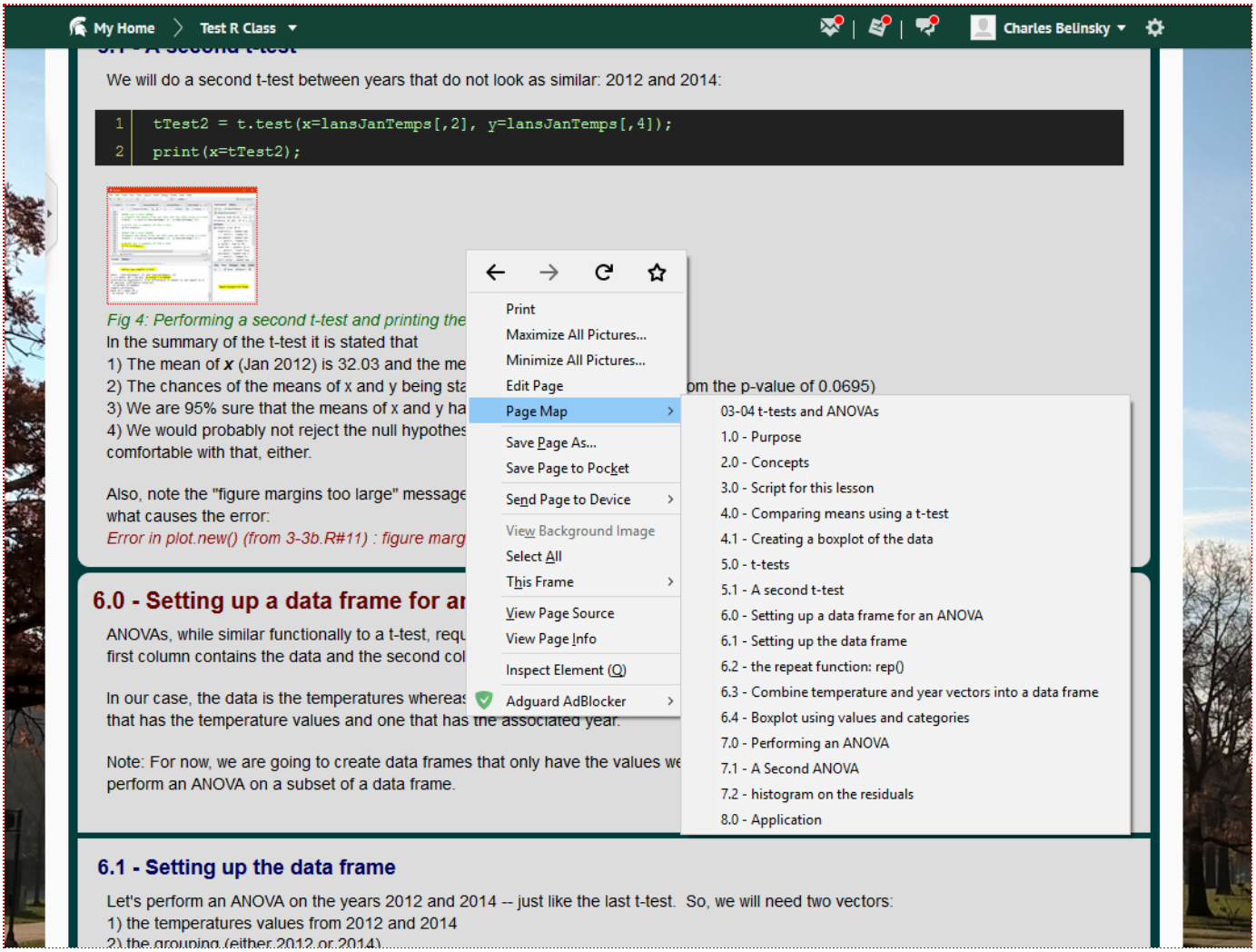

Fig 4: The right-click menu -- this view is from Firefox .

3.5 - Codeblocks

Double-clicking on a codeblock will select all the text in the codeblock -- you can then copy and paste it into RStudio.

```
# the next two lines should be at the top of all your scripts
rm(list=ls());
options(show.error.locations = TRUE);

# create three variables: d, t, and v
# give d and t values and use them to calculate v
d = 100;
t = 20;
v = d/t;
```

Fig 5: Double clicking on the codeblock selects all the text (code is from lesson 1-2).

Stand-alone code (i.e., can be copied to RStudio and executed as is) is colored blue (*Fig 5*).
Partial code is colored green (*Fig 6*).

```
# create three variables: d, t, and v
# give d and t values and use them to calculate v
d = 100;
t = 20;
v = d/t;
```

Fig 6: Partial code (i.e., meant to be added to a script -- not meant to be executed by itself).

4 - Personal statement from Charlie

The biggest thing that is lost when you move a class from a face-to-face environment to an online environment is the daily interaction between the instructor and the students. These interactions provide invaluable informal feedback for the instructor and, I would argue, are the main tool that an instructor uses to make improvements to their class. It is impossible to replicate this in an online class but I ask that you help me out and make an effort to communicate to me the little things. This could mean technical nags like content not appearing properly or pages loading too slowly, lesson content that is unclear, grammar and spelling issues, or scripts that does not work or work in a way that you do not understand.

Thank you for reading and taking this into consideration. In the end, it is the interactions between an instructor and the students that make a class great.

Now on to the stuff I have to put in a syllabus...

5 - Disclaimers and Loose Ends

5.1 - Student Responsibilities

- Students are expected to regularly check the ***Announcements*** section on the home page for the class and the email address associated with their MSU ID/ MSU Community ID for new information regarding the class.
- Students are expected to ask questions if they are having problems with an application or their class project.
- Students are expected to monitor their progress in order to complete the course work on time.
- Students are expected to contact either the instructor or MSU regarding technical issues that are interfering with the class.
- Students are expected to follow the MSU's academic integrity policy.
- Please notify the instructor regarding issues with the class website and lessons.

5.2 - Academic Integrity (summarized)

Written or other work which a student submits in a course, shall be the product of his/her own efforts. Plagiarism, cheating, and all other forms of academic dishonesty are prohibited. Students are expected to adhere to the ethical and professional standards associated with their programs and academic courses. All applicable portions of Michigan State's Policy on Academic Integrity apply to non-credit courses. Copies of the Policy on Academic Integrity may be accessed at https://www.msu.edu/unit/ombud/honestylinks.html.

5.3 - Instructor Responsibilities

- Instructor will be available for assistance for requested virtual office hours (through Zoom).
- Instructor will respond to emails from students within 1 business day. Potential for delayed responses (e.g., instructor is on vacation) will be posted on the front page of the course.
- Changes to the course will be posted in the ***Announcements*** section of the home page for the class.

5.4 - Attendance

You are expected to log in to the class at least once a week. Logging in allows you to stay updated and see new announcements.

5.5 - Participation

Please note that not all course lessons are the same length and the later ones tend to involve more work. Our goals with regard to participation/progress are to (a) keep you engaged in the course, (b) enhance the overall learning environment by promoting student-teacher communication, and (c) avoid a last minute time crunch for everyone involved. *We will not be sympathetic or make allowances for your failure on course tasks or deadlines that result from not seeing announcements because you had not accessed the course for an extended period, and had not told us you would be away from the internet.*

5.6 - ADA Statement

- MSU provides students with disabilities reasonable accommodations to participate in educational programs, activities, or services. Students with disabilities requiring accommodations to participate in class activities or meet course requirements should contact the instructor as early as possible.
- For students needing accommodations for disabilities, please contact your instructor and The Resource Center for Persons with Disabilities at Michigan State University at 517-353-9642.

6 - Trap: Bloatware

Adobe Acrobat, like most free software you download, attempts to bundle itself with software you most likely do not need nor want (i.e., bloatware). Make sure you take some time to read the optional offers whenever you download software. This author would argue that the main reason computers "slow" down over time is because of extra software that users unwittingly install.

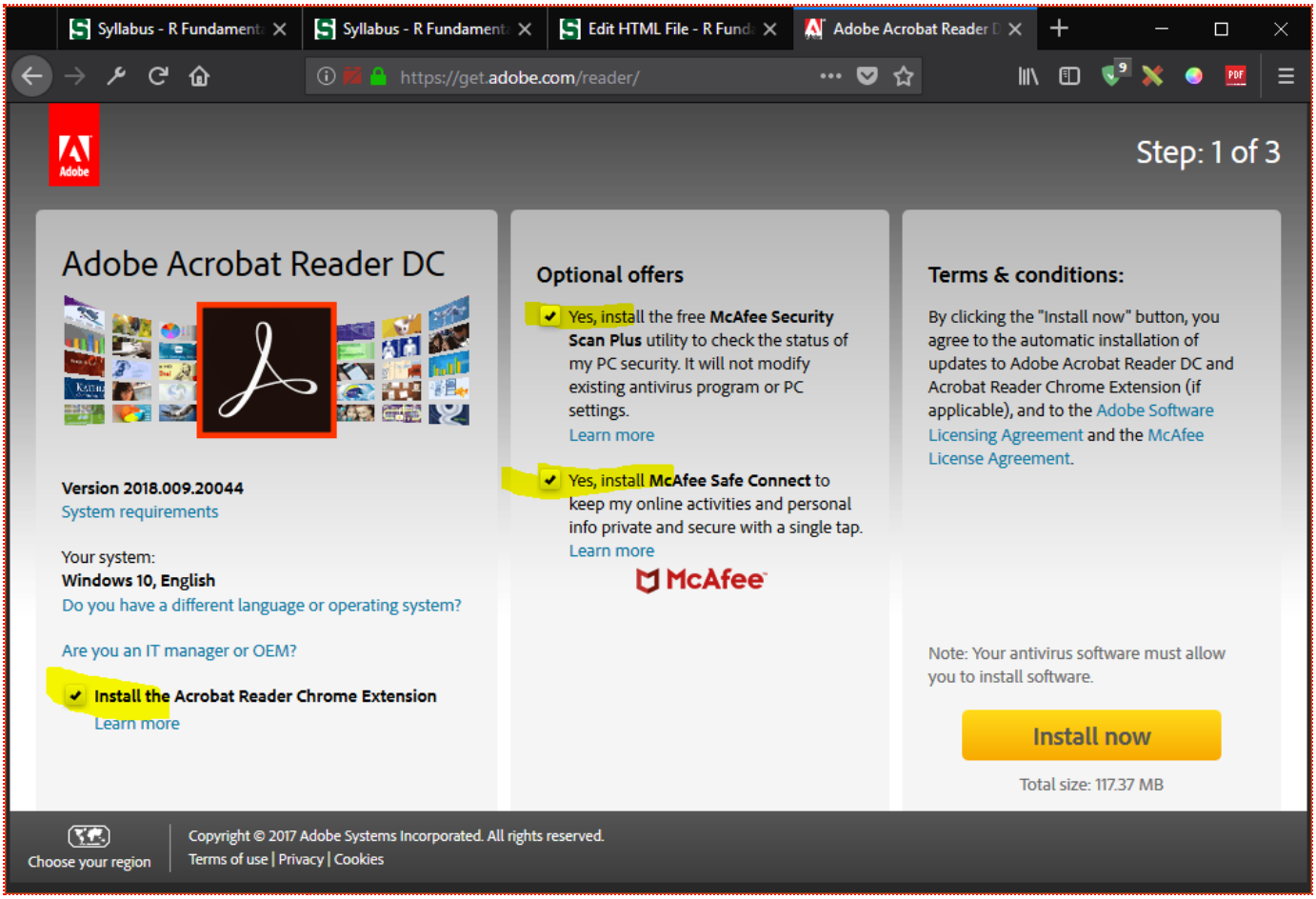

Fig 7: Three offers from Adobe that you neither need nor want -- this author recommends you uncheck them all.

7 - Extension: Testing the extension link.

Hi, and welcome to the test ***Extension***. If you clicked on the ***Extension*** link to get here then you can return to your previous location by right-clicking on this page and choosing ***Go to Previous Location***.

If you just scrolled down to this point then, congratulations, you have reached the end of the document.