LARSON—MATH 353–CLASSROOM WORKSHEET 07
Getting Started—Monty Hall Paradox.

1. Log in to CoCalc.

   (a) Start the Chrome browser.

   (b) Go to `https://cocalc.com`

   (c) Login (**your VCU email address** is probably your username).

   (d) You should see an existing Project for our class. Click on that.

   (e) Click "New", then "Worksheets", then call it **c07**.

2. **A problem to think about: the Monty Hall Paradox**. There are 3 doors to choose from (on say a game show). There is a car behind one door and a goat behind each of the other two. You choose one door (say, Door 1). Maybe it has the car behind it (or maybe not)? At least one of the other doors has a goat behind it—and one of these doors is opened (say, Door 2). You are then given an opportunity to switch your choice of door (in this example, you could switch from Door 1 to Door 3). Should you?

   How can you investigate this question experimentally? **On Wednesday we assumed that Door 0 was chosen. Modify Wednesday's code to investigate the case where Door 1 is chosen, and we switch our choice to the remaining door after the goat-door is revealed.**

Lists in Sage

A *list* is a basic *data structure* in Python and Sage. They are represented by square brackets with comma separated numbers, strings, etc., between them (like $[2, 5, 9]$ or `["red", "blue"]`). We have already seen lists in our use of both the `solve()` and `line()` commands which used, respectively, a list of equations and a list of points.

3. Lists can be given names. Evaluate `L=[2,3,5,9]`. Then evaluate `L`.

4. Lists are indexed starting with 0. Evaluate each of `L[0]`, `L[1]`, `L[2]`, and `L[3]`.

5. Lists can be combined with "+". Evaluate `[2,3,5,9]+[3,4,5]`. (Note: any common elements are repeated.)

6. Let `M=[3,4,5]`. Evaluate `L+M`.

7. If you want all the integers from x to y you can use the shorthand notation $[x..y]$. Evaluate `[3..7]`.

8. If you want a list with m n's you can use the shorthand notation $[n] * m$. Evaluate `[0]*7`.

9. You can have a list of lists. Evaluate `L=[[0,1],[2,3],[4,5]]`. Now evaluate `L[1]`. Then evaluate `L[1][0]`. What do you think the value of `L[0][1]` is?

10. You can use *list comprehension* to get a list of the values of any function applied to an initial list. Evaluate `[x**2 for x in [2,5,9]]`.

11. Use list comprehension to produce a list of the cubes of all the integers from 2 to 17.

12. List comprehension can also be used to *filter* the numbers in a list. Evaluate `[x for x in [2,5,9] if x%2==0]`. What did this do?

13. Evaluate `[x for x in [2,5,9] if x%2==1]`. What did this do?

    A list in Sage is a *mutable* object. Its entries can be changed.

14. Let L=[1, 2, 1, 2,1]. Evaluate $L[0]$ (you will get the 0th entry of L; it should be 1).

15. L[0] can be changed. Evaluate `L[0]=5`. Now evaluate L.

16. Let L = [2,3,3,3,2,1,8,6,3]. Try L.sort(). What does Sage do?

17. Try L.append(4). Evaluate L.

18. Try L.remove(3). What do you get?

19. Try sum(L). What do you get? What did Sage do?

20. Try prod(L). What do you get? What did Sage do?

21. Try [cos(x) for x in [0, pi/4, pi/2, 3*pi/4, pi]]. What do you get? Explain.

22. Try [factorial(x) for x in [1,2,3,4,5]]. What do you get? Explain?

23. What do you think the value of sum([exp(x) for x in [1,2,3,4,5]]) will be? Try it.

24. **Getting your classwork recorded**

    When you are done, before you leave class...

    (a) Click the "Make pdf" (Adobe symbol) icon and make a pdf of this worksheet. (If Cocalc hangs, click the printer icon, then "Open", then print or make a pdf using your browser).

    (b) Send me an email with an informative header like "Math 353—c07 worksheet attached" (so that it will be properly recorded).

    (c) Remember to attach today's classroom worksheet!