PLSC 31101: Computational Tools for Social Science

Rochelle Terman

Fall 2019

Contents

Part I

Before Class

Chapter 1

Syllabus

- Instructor: Rochelle Terman, rterman@uchicago.edu
- TA: Pete Cuppernull, pcuppernull@uchicago.edu
- Time: Tuesdays and Thursdays, 12:30 pm – 1:50 pm
- Place: Cobb Hall 303
- Office hours:
  - Rochelle Terman: Tuesdays, 2:30-4:30, Pick 411. Sign up here
  - Pete Cuppernull: Mondays, 2:00-4:00, Gates-Blake 211

1.1 Course Description

The purpose of this course is to provide graduate students with the critical computing skills necessary to conduct research in quantitative / computational social science. This course is not an introduction to statistics, computer science, or specialized social science methods. Rather, the focus will be on practical skills necessary to be successful in further methods work. The first portion of the class introduces students to basic computer literacy, terminologies, and the R programming language. The second part of the course provides students the opportunity to use the skills they learned in part 1 towards practical applications such as webscraping, data collection through APIs, automated text analysis, etc. We will assume no prior experience with programming or computer science.

Objectives

By the end of the course, students should be able to:

- Understand basic programming terminologies, structures, and conventions.
- Write, execute, and debug R code.
- Produce reproducible analyses using R Markdown.
- Clean, transform, and wrangle data using the `tidyverse` packages.
- Scrape data from websites and APIs.
- Parse and analyze text documents.
- Be familiar with the concepts and tools of a variety of computational social science applications.
- Master basic Git and GitHub workflows
- Learn independently and train themselves in a variety of computational applications and tasks through online documentation.

1.2 Who should take this course

This course is designed for Political Science Graduate students, but any graduate student is welcome. We will not presume any prior programming or computer science experience.

1.3 Requirements and Evaluation

Final Grades

This is a graded class based on the following:

- Completion of assigned homework (50%)
- Participation (25%)
- Final project (25%)

Assignments

Assignments will be assigned at the end of every Thursday session. They will be due one week later, unless otherwise noted. The assignments are intended to expand upon the lecture material and to help students develop the actual skills that will be useful for applied work. The assignments will be frequent but each of them should be fairly short.

You are encouraged to work in groups, but the work you turn in must be your own. Group submission of homework, or turning in copies of the same code or output, is not acceptable. While you are encouraged to use the internet to help you debug, but do not copy and paste large chunks of code that you do not understand. Remember, the only way you actually learn how to write code is to write code!

Portions of the homework in `R` should be completed using R markdown, a markup language for producing well-formatted documents with embedded R code and outputs. To submit your homework, knit the R Markdown file to PDF, and then submit the PDF file through Canvas (unless otherwise noted).

Class Participation

The class participation portion of the grade can be satisfied in one or more of the following ways:

- attending the lectures
- asking and answering questions in class
- attending office hours
- contributing to class discussion through the Canvas site, and/or
- collaborating with the computing community, either by attending a workshop or meetup, submitting a pull request to a github repository (including the class repository), answering a question on StackExchange, or other involvement in the social computing / digital humanities community.

Final Project

Students have two options for class projects:

1. **Data project**: Use the tools we learned in class on your own data of interest. Collect and/or clean the data, perform some analysis, and visualize the results. Post your reproducible code on Github.

2. **Tool project**: Create a tutorial on a tool we didn't cover in class. Ideas include: bash, LaTex, pandoc, quanteda, tidytext, etc. Post it on github.

Students are required to write a short proposal by **November 7** (no more than 2 paragraphs) in order to get approval / feedback from the instructors.

Project materials (i.e. a github repo) will be due by end of day on **December 9**. We will specify submission details in class.

On **December 10** we will have a **lightning talk session** where students can present their projects in a maximum 5 minute talk.

Late Policy and Incompletes

All deadlines are strict. Late assignments will be dropped a full letter grade for each 24 hours past the deadline. Exceptions will be made for students with a documented emergency or illness.

I will only consider granting incompletes to students under extreme personal/family duress.

Academic Integrity

I follow a zero-tolerance policy on all forms of academic dishonesty. All students are responsible for familiarizing themselves with, and following, university policies regarding proper student conduct. Being found guilty of academic dishonesty is a serious offense and may result in a failing grade for the assignment in question, and possibtire course.

1.4 Activities and Materials

Course Structure

Classes will follow a "workshop" style, combining lecture, demonstration, and coding exercises. We envision the class to be as interactive / hands on as possible, with students programming every session. **You must bring a laptop to class.**

It is important that students **complete the requisite reading** before class. I anticipate spending 1/2 the class lecturing, and 1/2 practicing with code challenges.

Course Website

All course notes will be posted on https://plsc-31101.github.io/course/, including class notes, code demonstrations, sample data, and assignments. Students will be assigned reading from these notes before every class.

Students are encouraged to submit pull requests to the website repository, for example if they find a typo, or if they found a particularly helpful resource that would aid other students.

Students projects will also be shared on the course website.

Canvas

We will use Canvas for communication (announcements and questions) and turning in assignments.

You should ask questions about class material and assignments through the Canvas website so that everyone can benefit from the discussion. We encourage you to respond to each other's questions as well. Questions of a personal nature can be emailed to us directly.

Computer Requirements and Software

By the end of the first week, you should install the following software on your computer:

- Access to the UNIX command line (e.g., a Mac laptop, a Bash wrapper on Windows)
- Git
- R and RStudio (latest versions)

This requires a computer that can handle all this software. Almost any Mac will do the job. Most Windows machines are fine too if they have enough space and memory.

See Install Page for more information. We will be having an **InstallFest on Wednesday, Oct 2 from 9:30 to 11:30 in Pick 504**. for those students experiencing difficulties downloading and installing the requisite software.

1.5 Curriculum Outline / Schedule

1. **Oct 1** - Introduction
    - course objectives, logistics, overview of programming and reproducible research.
2. **Oct 3** - R Tools
    - R Studio, R Markdown, packages, help.
3. **Oct 8** - R Syntax
    - basic syntax, variables, functions, data types.
4. **Oct 10** - Data Structures
    - vectors, lists, factors, matrices, data frames.
5. **Oct 15** - Indexing and Subsetting
    - subsetting formats, operators, boolean conditionals, sub-assignment, applications.
6. **Oct 17** - Introduction to Data
    - common terms, formats, tidy data, storage, import/export, exploratory data analysis.
7. **Oct 22** - Manipulating data with `dplyr`
    - importing data, subsetting, summarizing, and conducting calculations across groups, piping.

8. **Oct 24** - Tidying data with `tidyr`
    - tidy data principles, gather, spread, separate, unite
9. **Oct 29** - Merging and Linking Data
    - relational data, keys, joins, missing values.
10. **Oct 31** - Visualization
    - base plotting, ggplot, grammar of graphics, writing images.
11. **Nov 5** - Hypothesis testing and regressions
    - models, model objects, stargazer.
12. **Nov 7** - Strings and Regex
13. **Nov 12** - R programming 1
    - conditional flow, functions.
14. **Nov 14** - R Programming 2
    - iterations, map.
15. **Nov 19** - Collecting data with APIs
    - requests, RESTful APIs, queries, API libraries.
16. **Nov 21** - Webscraping
    - HTML, CSS, In-Browser tools, scraping tools.
17. **Nov 26** - Text analysis 1
    - preprocessing, DTM, dictionary methods, distinctive words.
18. **Nov 28** - No class, Thanksgiving
19. **Dec 3** - Text analysis 2
    - supervised vs. unsupervised learning, Vector-space models, topic models.
20. **Dec 5** - Git / Github
21. **Dec 10, 1:30-3:30pm** - Final project lightning talks

Chapter 2

Installation

To participate in PLSC 31101, you will need access to the software described below.

Once you've installed all of the software below, test your installation by following the instructions at the bottom on this page.

2.1 R

R is a programming language that is especially powerful for data exploration, visualization, and statistical analysis.

To download R, go to CRAN (the **C**omprehensive **R** **A**rchive **N**etwork).

A new major version of R comes out once a year, and there are 2-3 minor releases each year. It's a good idea to update regularly. Upgrading can be a bit of a hassle, especially for major versions, which require you to reinstall all your packages, but putting it off only makes it worse.

2.2 R Studio

To interact with R, we use RStudio. Please install the latest desktop version of RStudio IDE. We will not support RStudio cloud.

2.3 R Packages

You'll also need to install some R packages. An R package is a collection of functions, data, and documentation that extends the capabilities of base R. Using packages is key to the successful use of R.

Many of the packages that you will learn in this class are part of the so-called `tidyverse`. The packages in the `tidyverse` share a common philosophy of data and R programming, and are designed to work together naturally.

You can install the complete tidyverse with a single line of code in R Studio:

```
install.packages("tidyverse")
```

On your own computer, type that line of code in the RStudio console, and then press enter to run it. R will download the packages from CRAN and install them on to your computer. If you have problems installing, make sure that you are connected to the internet, and that https://cloud.r-project.org/ isn't blocked by your firewall or proxy.

We will also be requiring a few other important packages. Run the following line of code to download all of them at once:

```
install.packages("rmarkdown", "knitr", "gapminder",
                 "stargazer", "rtweet", "kableExtra",
                 "nycflights13", "devtools", "tm", "wordcloud",
                 "matrixStats", "SnowballC", "tidytext",
                 "textdata", "stm")
```

2.4 LaTex

In order to knit rmarkdown files to pdf files, you need to install some version of Latex. For students who have not installed LaTeX before, we recommend that you install TinyTeX (https://yihui.name/tinytex/).

Open RStudio and type these lines into the command-line console:

```
install.packages("tinytex")
tinytex::install_tinytex()
```

2.5 The Bash Shell

Bash is a commonly-used shell that gives you the power to do simple tasks more quickly.

Windows

Install Git for Windows by downloading and running the installer. This will provide you with both Git and Bash in the Git Bash program. **NOTE**: on the ~6th step of installation, you will need to select the option "Use Windows' default console window" rather than the default of "Use MinTTY" in order for nano to work correctly.

After the installer does its thing, it leaves the window open, so that you can play with the "Git Bash".

Chances are that you want to have an easy way to restart that Git Bash. You can install shortcuts in the start menu, on the desktop or in the QuickStart bar by calling the script /share/msysGit/add-shortcut.tcl (call it without parameters to see a short help text).

Mac OS X

The default shell in all versions of Mac OS X is bash, so no need to install anything. You access bash from the Terminal (found in `/Applications/Utilities`). You may want to keep Terminal in your dock for this class.

Linux

The default shell is usually Bash, but if your machine is set up differently you can run it by opening a terminal and typing bash. There is no need to install anything.

2.6 Git

Git is a version control system that lets you track who made changes to what when and has options for easily updating a shared or public version of your code on github.com. You will need a supported web browser (current versions of Chrome, Firefox or Safari, or Internet Explorer version 9 or above).

Windows

Git should be installed on your computer as part of your Bash install (described above).

Mac OS X

For OS X 10.9 and higher, install Git for Mac by downloading and running the most recent "mavericks" installer from this list. After installing Git, there will not be anything in your `/Applications` folder, as Git is a command line program. **For older versions of OS X (10.5-10.8)** use the most recent available installer labelled "snow-leopard" available here.

Linux

If Git is not already available on your machine you can try to install it via your distro's package manager. For Debian/Ubuntu run sudo apt-get install git and for Fedora run sudo yum install git.

2.7 Other helpful tools

While not required, I recommend you install the following tools:

1. Google Chrome is an up-to-date web browser.
2. Sublime Text is a free, advanced text editor.

2.8 Testing your installation

If you have trouble with installation, please come to the Installfest on **Wed Oct 2, 10-12 in Pick 411**.

Open a command line window ('terminal' or, on windows, 'git bash'), and enter the following commands (without the $ sign):

```
$ R --version
$ git --version
```

If everything has been installed correctly, those commands *should* print output version information.

NB: If you're using git bash, the `R --version` command may not work. In this case, just make sure you can open up RStudio.

Software Carpentry maintains a list of common issues that occur during installation may be useful for our class here: Configuration Problems and Solutions wiki page.

Credit: Thanks to Software Carpentry for providing installation guidelines.

Chapter 3

Homework Rubric

Each question will be graded along the following criteria:

Check Minus: The code doesn't run, or does not do what the question is asking.

Check: The code works, but exhibits one of the following problems:

1) poor documentation / disorganized;
2) repetitive / redundant / overly complex;
3) graphs and tables lack labels or are otherwise difficult to understand.

Check Plus: The code works, and is:

1) well-documented and organized;
2) clean and efficient;
3) graphs and tables have proper labels and are easily readable.

Part II

Course Notes

Chapter 4

Introduction

4.1 The Motivation

Here's the dream:

....

> Computers have revolutionized research, and that revolution is only beginning. Every day, social scientists and humanists all over the world use them to study things that are too big, too small, too fast, too slow, too expensive, too dangerous, or just too hard to study any other way.

Now here's the reality

....

> Every day, scholars all over the world waste time wrestling with computers. Tasks that should take a few moments take hours or days, and many things never work at all. When scholars try to get help, they are inundated with unhelpful information, and give up.

This sorry state of affairs persists for three reasons:

1. **No room, no time.** Everybody's schedule is full —- there's simply not space to add more about computing without dropping something else.

2. **The blind leading the blind.** The infrastructure doesn't exist to help scholars develop the skills they need. Senior researchers can't teach the next generation how to do things that they don't know how to do themselves.

3. **Autodidact Chauvinism.** Since there are no classrooms, scholars are pressured to teach themselves. But self-learning is time consuming and

nearly impossible without a base level of knowledge.

Despite these challenges, there are great reasons to learn how to program:

1. Practical Efficiency

Even though it takes some time at first, learning how to program can save you an enormous amount of time doing basic tasks that you would otherwise do by hand, once you get the hang of it.

2. New Tools

Some things are impossible, or nearly impossible to do by hand. Computers open the door for new tools and methods, but many require programming skills.

3. New Data

The Internet is a wealth of data, waiting to be analyzed! Whether its collecting Twitter data, working with the Congress API, or scraping websites, programming knowledge is a must.

4. Better Scholarship

(Quality) programming can open the door to better transparency, reproducibility, and collaboration in the Social Sciences and the Humanities.

Goal of the Class: Learn to Learn

> The basic learning objective of this course is to leave here with the knowledge and skills required to learn on your own, whether that's through programming documentation, StackExchange and other online fora, courses or D-Lab workshops.

By the end of the course, students should be able to:

- Understand basic programming terminologies, structures, and conventions.
- Write, execute, and debug R code.
- Produce reproducible analyses using R Markdown.
- Clean, transform, and wrangle data using the `tidyverse` packages.
- Scrape data from websites and APIs.
- Parse and analyze text documents.
- Be familiar with the concepts and tools of a variety of computational social science applications.
- Master basic Git and GitHub workflows
- **Learn independently and train themselves in a variety of computational applications and tasks through online documentation.**

This Course will not…

- teach you to be a professional programmer or software developer

- teach you statistics, computer science, or specialized social science / digital humanities methods

Why not just take a Computer Science course?

Computer science courses don't anticipate the types of questions social scientists might ask, and therefore they - introduce many unnecessary concepts - do a poor job of explaining how computer programming tools might be used by social scientists - are too resource intensive for the average social scientist

Programming is not just use useful for computer scientists, methodologists, or people who work with "big data."

A Practical example

To illustrate, here's a practical example that comes out of my own research:

Task 1 (by hand)

Topic: International Human Rights "Naming and Shaming"

Question: Who Shames Whom on What?

Data: UN Human Rights Committee's Universal Periodic Review

From Antigua & Barbuda 2011 review:

1. Continue with the efforts to prevent, punish and eradicate all forms of violence against women (Argentina);
2. Accede to the Second Optional Protocol to the International Covenant on Civil and Political Rights, aimed at abolishing the death penalty, and take all necessary steps to remove the death penalty from Antigua and Barbuda law (Australia);
3. Improve conditions in Antigua and Barbuda's prisons and detention facilities (Australia);

The Task: Parse a bunch of reports (PDFs) into a dataset (CSV). Add metadata for issue, action, sentiment.

From	To	Text	Action	Sentiment	Issue	Institution	Response
Argentina	Antigua & Barbuda	Continue with the efforts to prevent, punish and eradicate all forms of violence against women	continue	2	Women's rights		support
Australia	Antigua & Barbuda	Accede to the Second Optional Protocol to the International Covenant on Civil and Political Rights, aimed at abolishing the death penalty, and take all necessary steps to remove the death penalty from Antigua and Barbuda law	accede	5	Death penalty,International instruments	ICCPR, OP	reject
Australia	Antigua & Barbuda	Improve conditions in Antigua and Barbuda's prisons and detention facilities	improve	4	Detention		support

How much time will it take?

By hand: 40,000 recommendations x 3 min per recommendation x 8-hour days x 5-day weeks = **12 months**

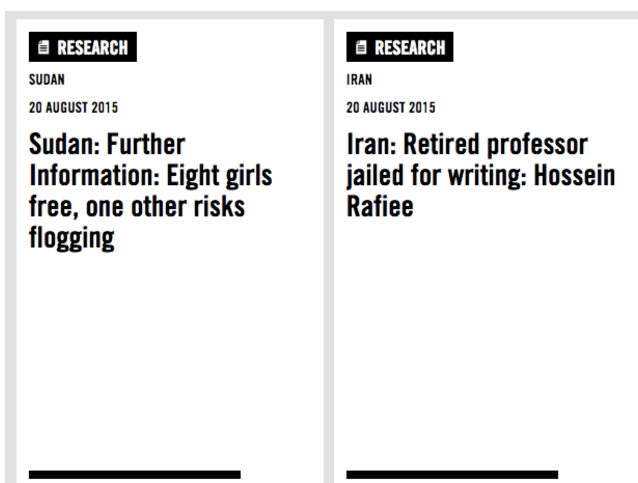
Task 2 (by hand)

What if we wanted to extend this research?

Question: How does UPR shaming compare to actual human rights abuses?

Data: Amnesty International's Urgent Actions

Task: Collect all of Amnesty International's Urgent Actions, add metadata for issue, country.



How much time will it take?

By hand: 25,000 recommendations x 3 min per recommendation x 8-hour days x 5-day weeks = **7.5 months**

Tasks 1 & 2 (with a computer)

With a computer, we can write a program that:

1. Parses recommendations into a CSV.
2. Codes recommendations by issue, action, sentiment using computational text analysis tools.
3. Uses webscraping to collect all of Amnesty International's urgent actions.
4. Run simple regression models with R to correlate Amnesty reports with UPR shaming.

Total time: 2 months

Time Saved: 1.5 years

4.2 About This Class

About Me

My name is Rochelle Terman and I'm a faculty member in Political Science.

- A few years ago, I didn't know how to program. Now I program almost every day.
- I program mostly in Python and R. I have a special interest in text analysis and webscraping.
- My substantive research is on international norms and human rights.
- I won't be able to answer all your questions.
- No one will.
- But especially me.

Course Structure

The course is divided into two main sections:

1. Skills

Basic computer literacy, terminologies, and programming languages:

- Base R: objects and data structures.
- `tidyverse` for data analysis.
- Modeling and visualization.
- Key programming concepts (iteration, functions, conditional flow, etc).

We're using `R` because it's the common programming language among Political Scientists. But if you understand the *concepts*, you should able to pick up Python and other languages pretty easily.

2. Applications

Use the skills they learned in part 1 towards practical applications:

- Webscraping.
- APIs.
- Computational Text Analysis.
- Version control and communication.

The goal is to **introduce** the students to a medley of common applications so that they can discover which avenue to pursue in their own research, and what such training would entail.

Class Activities

Classes will follow a "workshop" style, combining lecture, demonstration, and coding exercises. We envision the class to be as interactive / hands on as possible, with students programming every session. **You must bring a laptop to class.**

It is important that students **complete the requisite reading** before class. I anticipate spending 1/2 the class lecturing, and 1/2 practicing with code challenges.

Course Websites

Class notes and other materials are available here: https://github.com/plsc-31101/course/

We will also be using Canvas to distribute/accept assignments, and for discussion. Please use the discussion forums liberally.

Evaluation

This is a graded class based on the following: - Completion of assigned homework (50%) - Participation (25%) - Final project (25%)

If you want to audit, please let me know ASAP.

Assignments

- In general, assignments are assigned at the end of lecture, and due the following week.
- Exceptions will be noted.
- The first assignment is due next week, before class on Thursday Oct 10. It is on Canvas.
- Turn in assignments on Canvas.
- Work in groups, but submit your own.

Participation

The class participation portion of the grade can be satisfied in one or more of the following ways:

- attending the lectures,
- asking and answering questions in class,
- attending office hours,
- contributing to class discussion through the Canvas/Piazza site,
- collaborating with the computing community.

Final Project

Students have two options for class projects:

1. *Data project*: Using the tools we learned in class on your own data of interest.

2. *Tutorial project*: Create a tutorial on a tool we didn't cover in class.

Both options require an R markdown file narrating the project.

Students are required to write a short proposal by **November 7** (no more than 2 paragraphs) in order to get approval / feedback from the instructors.

Project materials (i.e. a github repo) will be due by end of day on **December 9**. We will specify submission details in class.

On **December 10** we will have a **lightning talk session** where students can present their projects in a maximum 5 minute talk.

Software

- Installation instructions are onthe website.
- Get started **EARLY**.
- Then go to the Installfest (Wednesday, Oct 2, 9:30-11:30 in Pick 504) to double check your installation.
- If you have computer troubles, post the problem on the discussion forums, with as much detail as possible.

4.3 Learning How to Program

Before we talk about what it takes to learn how to program, let's first review what programming is.

What is Programming?

A *program* is a sequence of instructions that specifies how to perform a computation. Most programs are written in a human-readable *programming language* (or "source code") and then executed with the help of a *compiler* or *interpreter*.

A few basic instructions appear in just about every language:

1. **input**: Get data from the keyboard, a file, the network, or some other device.
2. **output**: Display data on the screen, save it in a file, send it over the network, etc.
3. **math**: Perform basic mathematical operations like addition and multiplication.
4. **conditional execution**: Only perform tasks if certain conditions are met.
5. **iteration**: Do the same task over and over again on different inputs.

That being said, programming languages differ from one another in the following ways:

1. **Syntax**: whether to add a semicolon at the end of each line, etc.
2. **Usage**: JavaScript is for building websites, R is for statistics, Python is general purpose, etc.
3. **Level**: how close you are to the hardware. 'C' is often considered to be the lowest (or one of the lowest) level languages.
4. **Object-oriented:** "objects" are data + functions. Some programming languages are object-oriented (e.g. Python) and some aren't (e.g. C).
5. **Many more**: Here's a list of all the types of programming languages out there.

What Language Should You Learn?

Most programmers can program in more than one language. That's because they know *how to program* generally, as opposed to "knowing" Python, R, Ruby, or whatever.

So what should your first programming language be? That is, what programming language should you use to learn *how to program*? At the end of the day, the answer depends on what you want to get out of programming. Many people recommend Python because it's fun, easy, and multi-purpose. Here's an article that can offer more advice.

In this class, we'll be using R because it's the most popular language in our disciplnary community (of Political Scientists.)

Regardless of what you choose, you will probably grow comfortable in one language while learning the basic concepts and skills that allow you to 'hack' your way into other languages. That's because **programming is an extendible skill.**

Thus "knowing how to program" means learning how to *think* like a programmer, not necessarily knowing all the language-specific commands off the top of your head. **Don't learn specific programming languages; learn how to program.**

What Programming is Really Like

Here's the sad reality: When you're programming, 80% or more of your time will be spent debugging, looking stuff up (like program-specific syntax, documentation for packages, useful functions, etc.), or testing. This does not just apply to beginner or intermediate programmers, although you will grow more "fluent" over time.

> Google software engineers write an average of 10-20 lines of code per day. **The Lesson**: Programming is a slow activity, especially in the beginning.

If you're a good programmer, you're a good detective!

How to Learn

Here are some tips on how to learn computer programming:

1. Learning to program is 5% intelligence, 95% endurance.
2. Like learning to play an instrument or speak a foreign language, it takes practice, practice, practice.
3. Program a little bit every day.
4. Program with others. Do the problem sets in pairs or groups.
5. It's better to type than to copy and paste.

6. Most "programming" is actually researching, experimenting, thinking.
7. Stay organized.

The 15 minute rule

15 min rule: when stuck, you HAVE to try on your own for 15 min; after 15 min, you HAVE to ask for help.- Brain AMA pic.twitter.com/MS7FnjXoGH

— Rachel Thomas (?) August 14, 2016

We will follow the **15 minute rule** in this class. If you encounter a problem in your assignments, spend 15 minutes troubleshooting the problem on your own. After 15 minutes, if you still cannot solve the problem, **ask for help**.

(Hat tip to Computing for Social Sciences)

Debugging

Those first 15 minutes should be spent trying to debug your code. Here are some tips:

- Read the errors!
- Read the documentation
- Make it smaller
- Figure out what changed
- Check your syntax
- Print statements are your friend

Using the Internet

You should also make use of Google and StackOverflow to resolve the error. Here's some tips for how to google errors:

- google: name-of-program + text in error message.
- Remove user- and data-specific information first!
- See if you can find examples that do and don't produce the error. Try other people's code, but don't fall into the copy-paste trap.

Asking for Help

We will use Canvas/Piazza for class-related questions and discussion. You are highly encouraged to ask questions, and answer one another's questions.

1. Include a brief, informative title.
2. Explain what you're trying to do, and how it failed.

3. Include a reproducible example.

Here are some helpful guidelines on how to properly ask programming questions:

1. "How to Ask Programming Questions," ProPublica
2. "How do I ask a good question?" StackOverflow
3. "How to properly ask for help" Computing for Social Science

Chapter 5

R Basics

This unit introduces you to the R programming language, and the tools we use to program in R. We will explore:

1. **What is R?**, a brief introduction to the R language;
2. **R Studio**, a tour of the Interactive Development Environment RStudio;
3. **R Packages**, extra tools and functionalities;
4. **R Markdown**, a type of R script file we'll be working with in this class.

5.1 What is R?

R is a versatile, open source programming and scripting language that's useful both for statistics and data science. It's inspired by the programming language `S`. Some of its **best features** are:

- It's free, open source, and available on every major platform. As a result, if you do your analysis in R, most people can easily replicate it.
- It contains massive set of packages for statistical modelling, machine learning, visualisation, and importing and manipulating data. Over 14,000 packages are available as of August 2019. Whatever model or graphic you're trying to do, chances are that someone has already tried to do it (and a package for it exists).
- It's designed for statistics and data analysis, but also general-purpose programming.
- It's an Interactive Development Environment tailored to the needs of interactive data analysis and statistical programming.
- It has powerful tools for communicating your results. R packages make it easy to produce html or pdf reports, or create interactive websites.

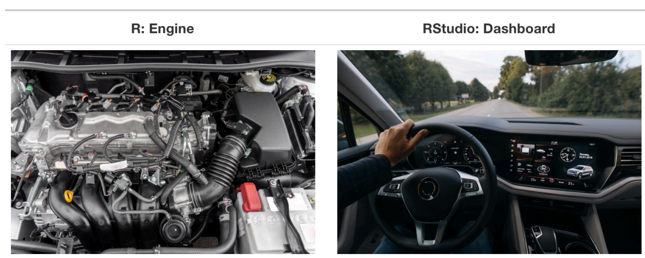
- A large and growing community of peers.

R also has a number of **shortcomings**:

- It has a steeper learning curve than SPSS or Stata.
- R is not a particularly fast programming language, and poorly written R code can be terribly slow. R is also a profligate user of memory.
- Much of the R code you'll see in the wild is written in haste to solve a pressing problem. As a result, code is not very elegant, fast, or easy to understand. Most users do not revise their code to address these shortcomings.
- Inconsistency is rife across contributed packages, even within base R. You are confronted with over 20 years of evolution every time you use R. Learning R can be tough because there are many special cases to remember.

5.2 R Studio

Throughout this class, we will assume that you are using R via RStudio. First time users often confuse the two. At its simplest R is like a car's engine while RStudio is like a car's dashboard.



More precisely, R is a programming language that runs computations, while RStudio is an *integrated development environment (IDE)* that provides an interface with many convenient features and tools. So just as the way of having access to a speedometer, rear-view mirrors, and a navigation system makes driving much easier, using RStudio's interface makes using R much easier as well.

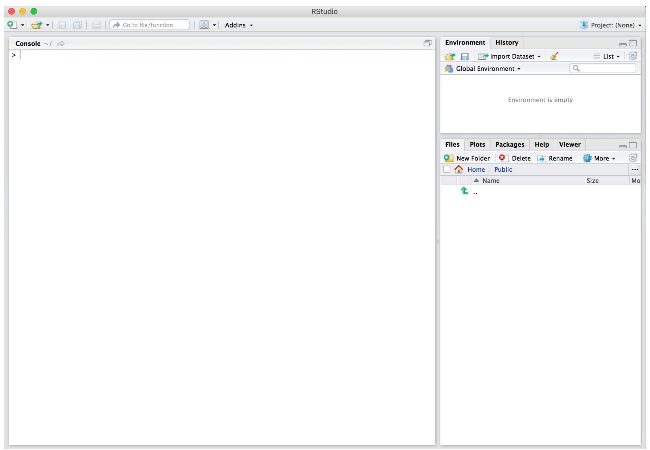
RStudio includes a console, syntax-highlighting code editor, as well as tools for plotting, history, debugging and workspace management. It's also free and open-source. Yay!

> NB: We don't have to use RStudio to use R. For example, we write R code in a plain text editor (like `textedit` or `notepad`), and then execute the sript using the shell (e.g. `terminal` in Mac). But this is not ideal.

After you install R and RStudio on your computer, you'll have two new applications you can open. We'll always work in RStudio– not in the R application.

R: Do not open this	RStudio: Open this

After you open RStudio, you should see something similar to this:



5.2.1 Console

There are two main ways of interacting with R: using the **console** or by using the **script editor**.

The console window (in RStudio, the bottom left panel) is the place where R is waiting for you to tell it what to do, and where it will show the results of a command.

You can type commands directly into the console, but they will be forgotten when you close the session. Try it out now.

```
> 2 + 2
```

If R is ready to accept commands, the R console shows a `>` prompt. If it receives a command (by typing, copy-pasting or sent from the script editor using `Ctrl-Enter`), R will try to execute it, and when ready, show the results and come back with a new `>`-prompt to wait for new commands.

If R is still waiting for you to enter more data because it isn't complete yet, the console will show a `+` prompt. It means that you haven't finished entering a complete command. This happens when you have not 'closed' a parenthesis or quotation. If you're in RStudio and this happens, click inside the console window and press `Esc`; this should help get you out of trouble.

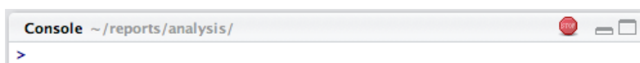
```
> "This is an incomplete quote
+
```

More Console Features

1. Retrieving previous commands: As you work with R you'll often want to re-execute a command which you previously entered. Recall previous commands using the up and down arrow keys.

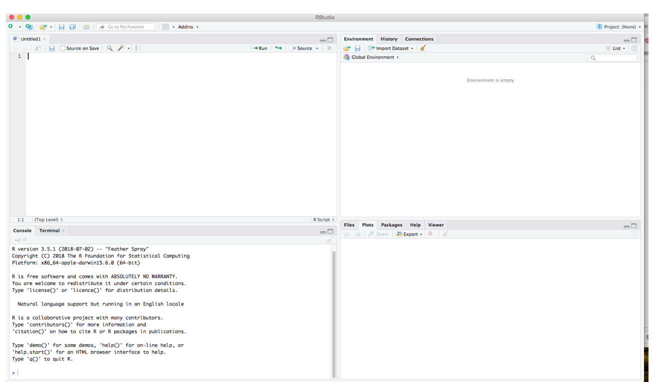
2. Console title bar: This screenshot illustrates a few additional capabilities provided by the Console title bar:

   - Display of the current working directory.
   - The ability to interrupt R during a long computation.
   - Minimizing and maximizing the Console in relation to the Source pane (using the buttons at the top-right or by double-clicking the title bar).



5.2.2 Scripts

It's better practice to enter commands in the script editor, and save the script. This way, you have a complete record of what you did, you can easily show others how you did it and you can do it again later on if needed. Open it up either by clicking the *File* menu, and selecting *New File*, then R script, or using the keyboard shortcut Cmd/Ctrl + Shift + N. Now you'll see four panes.



The script editor is a great place to put code you care about. Keep experimenting in the console, but once you have written code that works and does what you want, put it in the script editor.

RStudio will automatically save the contents of the editor when you quit RStudio, and will automatically load it when you re-open. Nevertheless, it's a good idea to save your scripts regularly and to back them up.

5.2.3 Running code

While you certainly can copy-paste code you'd like to run from the editor into the console, this workflow is pretty inefficient. The key to using the script editor effectively is to memorize one of the most important keyboard shortcuts in RStudio: **`Cmd/Ctrl + Enter`**. This executes the current R expression from the script editor in the console.

For example, take the code below. If your cursor is somewhere on the first line, pressing `Cmd/Ctrl + Enter` will run the complete command that generates `dems`. It will also move the cursor to the next statement (beginning with `reps`). That makes it easy to run your complete script by repeatedly pressing `Cmd/Ctrl + Enter`.

```
dems <- (55 + 70) * 1.3

reps <- (20 - 1) / 2
```

Instead of running expression-by-expression, you can also execute the complete script in one step: `Cmd/Ctrl + Shift + S`. Doing this regularly is a great way to check that you've captured all the important parts of your code in the script.

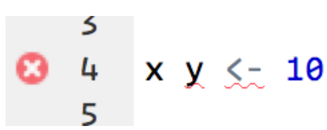
5.2.4 Comments

Use `#` signs to add comments within your code chunks, and you are encouraged to regularly comment within your code. Anything to the right of a `#` is ignored by R. Each line of a comment needs to begin with a #.

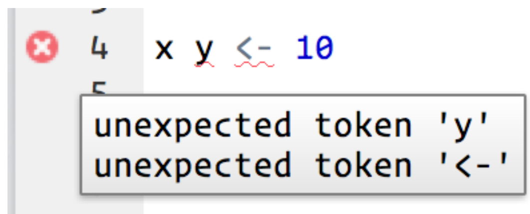
```
# This is a comment.
# This is another line of comments.
```

5.2.5 Diagnostics and errors

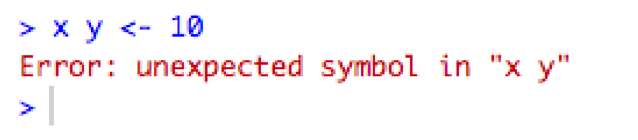
The script editor will also highlight syntax errors with a red squiggly line and a cross in the sidebar:



You can hover over the cross to see what the problem is:



If you try to execute the code, you'll see an error in the console.



When errors happen, your code is haulted – meaning it's never executed. Errors can be frustrating in R, but with practice you'll be able to debug your code quickly.

5.2.6 Errors, Messages, and Warnings

One thing that intimidates new R and RStudio users is how it reports errors, warnings, and messages. R reports errors, warnings, and messages in a glaring font, which makes it seem like it is scolding you. However, seeing red text in the console is not always bad.

1. **Errors**: When the text is a legitimate error, it will be prefaced with "Error:" and try to explain what went wrong. Generally when there's an error, the code will not run. *Think of errors as a red traffic light: something is wrong!*

2. **Warnings**: When the text is a warning, it will be prefaced with "Warning:" and R will try to explain why there's a warning. Generally your code will still work, but perhaps not in the way you would expect. *Think of warnings as a yellow traffic light: everything is working fine, but watch out/pay attention.*

3. **Messages**: When the text doesn't start with either "Error" or "Warning", it's just a friendly message. These are helpful diagnostic messages and they don't stop your code from working. *Think of messages as a green traffic light: everything is working fine and keep on going!*

8XkXd _ 1MpB`QMK2Mi

hm`M vQm` ii2MiBQM iQ i?2 mTT2` `B;?i T M2X h?Bb T M2 /BbTH vb vQm
2MpB`QMK2Miô- M/ Bi +QMi BMb i?2 / i Q#D2+ib vQm ? p2 b p2/ BM vQm
b2bbBQMX LQiB+2 i? i r2 ? p2 i?2 irQ Q#D2+ib M/ 2 T2 i2/ 2 `HB2`-
HQM;rBi? i?2B` p Hm2bX

uQm + M HBbi HH Q#D2+ib BM vQm` +m``2Mi 2MpB`QMK2Mi #v `mMMBM;,

```
Hb UV
O= (R) ]/2Kb] ]`2Tb]
```

aQK2iBK2b r2 r Mi iQ `2KQp2 Q#D2+ib i? i r2 MQ HQM;2` M22/X

```
t I@8
`KUtV
```

A7 vQm r Mi iQ `2KQp2 HH Q#D2+ib 7`QK vQm` +m``2Mi 2MpB`QMK2Mi- vQ

```
`KUHBbi 4 HbUVV
```

+FMQrH2/;K2Mib

h?Bb T ;2 Bb BM T `i /2`Bp2/ 7`QK i?2 7QHHQrBM; bQm`+2b,

RX_ aim/BQ amXTTQ`i

kX/p M+2/-_HB+2Mb2/ mM/2` i?2 *QKKQMb ii`B#miBQM@
LQM*QKK2`+B H@a? `2 HBF2 9Xy AMi2`M iBQM H Sm#HB+ GB+2Mb2

jX_ 7Q` . i a+B2M+2 mM/2` i?2 *QKKQMb ii`B#miBQM@
LQM*QKK2`+B H@LQ.2`Bpb jXy

9X JQ/2`M.Bp2 BMiQ _ M/ i?2 iB/vp2`b2

8Xj _ S +F ;2b

h?2 #2bi T `i #Qmi _ `2 Bib mb2`@+QMi`B#mi2/ T +F ;2b U HbQ + HH2/ óH
T +F ;2b +QHH2+iBQM Q7 7mM+iBQMb U M/ bQK2iBK2b / i V i? i + M #2
Qi?2` T`Q;` K2`bX

;QQ/ M HQ;v 7Q` _ T +F ;2b Bb i?2v `2 HBF2 TTb vQm + M /QrMHQ / Q
KQ#BH2 T?QM2,

aQ_ Bb HBF2 M2r KQ#BH2 T?QM2, r?BH2 Bi ? b +2`i BM KQmMi
vQm mb2 Bi 7Q` i?2 }`bi iBK2- Bi /Q2bMöi ? p2 2p2`vi?BM;X _T +F
TTb vQm + M /QrMHQ / QMiQ vQm` T?QM2 7`QK TTH2öb TT aiQ`
:QQ;H2 SH vX

G2iöb +QMiBMm2 i?Bb M HQ;v #v +QMbB/2`BM; i?2 AMbi ;` K TT
b? `BM; TB+im`2bX a v vQm ? p2 Tm`+? b2/ M2r T?QM2 M/ vQm
b? `2 T?QiQ vQm ? p2 Dmbi i F2M rBi? 7`B2M/b M/ 7 KBHv QM A
M22/ iQ,

RX AMbi HH i?2 aTT M+2 vQm` T?QM2 Bb M2r M/ /Q2b MQi BM+H
AMbi ;` K TT- vQm M22/ iQ /QrMHQ / i?2 TT 7`QK 2Bi?2` i?2
Q` :QQ;H2 SH vX uQm /Q i?Bb QM+2 M/ vQmö`2 b2i 7Q` i?2 iB
KB;?i M22/ iQ /Q i?Bb ; BM BM i?2 7mim`2 r?2M i?2`2 Bb M m
TTX

kX PT2M i?2 TT ,7i2` vQmöp2 BMbi HH2/ AMbi ;` K- vQm M22/ iQ Q
TTX

h?2 T`Q+2bb Bb p2`v bBKBH ` 7Q` mbBM; M _T +F ;2X uQm M22/

RX AMbi HH i?2 T +F ;2 ,h?Bb Bb HBF2 BMbi HHBM; M TT QM vQm` T?Q
T +F ;2b `2 MQi BMbi HH2/ #v /27 mHi r?2M vQm BMbi HH _
h?mb B7 vQm r Mi iQ mb2 T +F ;2 7Q` i?2 }`bi iBK2- vQm M22
Bi }`biX PM+2 vQmöp2 BMbi HH2/ T +F ;2- vQm HBF2Hv rQMö
mMH2bb vQm r Mi iQ mT/ i2 Bi iQ M2r2` p2`bBQMX

kX óGQ /ô i?2 T +F ;2 ,GQ /BM; T +F ;2 Bb HBF2 QT2MBM; M TT QM
T?QM2X S +F ;2b `2 MQi HQ /2/ #v /27 mHi r?2M vQm bi `i _
vQm` +QKTmi2`c vQm M22/ iQ HQ / 2 +? T +F ;2 vQm r Mi iQ mb
vQm bi `i _aim/BQX

8XjXR AMbi HHBM; S +F ;2b

6B`bi- r2 /QrMHQ / i?2 T +F ;2 7`QK QM2 Q7 i?2 *_ L KB``Q`b QMi
Tmi2`X 6Q` i?Bb HBM2 Q7 +Q/2X T +F ;2bU]T +F ;2@M K2]V p2 MQi
b2i T`272``2/ *_ L KB``Q` QT2MbX Q r BM/Qr rBHH TQT mT bFBM;
vQm iQ +?QQb2 HQ+ iBQMX

G2iöb /QrMHQ / i?2 TTHvX`*F ;2

```
BMbi HHXT +F ;2 bV [UFT;Hv]
```

A7 vQm `mM BMiQ 2``Q`b H i2` BM i?2 +Qm`b2 #Qmi 7mM+iBQM Q` T +F 7QmM/- `mM i?2 BMbi HHXT +F ;2 7mM+iBQM iQ K F2 bm`2 i?2 T +F ;2 Bb +im HHv BMbi HH2/X

AKTQ`i,MiPM+2 r2 /QrMHQ / i?2 T +F ;2- r2 M2p2` M22/ iQ `mM BMbi HHXT +F ;2 bUmmH2bb r2 ;2i M2r +QKTmi2`XV

8XjXk GQ /BM; S +F ;2b

PM+2 r2 /QrMHQ / i?2 T +F ;2- r2 M22/ iQ HQ / Bi BMiQ Qm` b2bbBQM iQ mb2 Bb `2[mB`2/ i i?2 #2;BMMBM; Q7 2 +? _ b2bbBQMX h?Bb bi2T Bb M2+2bb r2 miQK iB+ HHv HQ /2/ 2p2`v T +F ;2 r2 ? p2 2p2` /QrMHQ /2/- Qm` +QKT rQmH/ 7`vX

```
HB#` `vUTHv`V
```

h?2 K2bb ;2 i2HHb vQm r?B+? T7mM+iBQMb 7`QK7mM+iBQMb BM # b2 _ UQ` 7`QK Qi?2` T +F ;2b vQm KB;?i ? p2 HQ /2/VX

8XjXj *? HH2M;2b

G2iöb ;Q ?2 / M/ /QrMHQ / bQK2 +Q`2- BKTQ`i Mi T +F ;2b r2öHH mb2 7 `2bi Q7 i?2 +Qm`b2X .QrMHQ / UB7 vQm ? p2Möi /QM2 bQ H`2 /vV M/ H 7QHHQrBM; T +F ;2b,

- Ç iB/vp2`b2
- Ç `K `F/QrM
- Ç FMBi`
- Ç ; TKBM/2`
- Ç /2piQQHb
- Ç bi `; x2`
- Ç `ir22i
- Ç F #H21ti`

+FMQrH2/;K2Mib

h?Bb T ;2 Bb BM T `i /2`Bp2/ 7`QK i?2 7QHHQrBM; bQm`+2b,

RX_ aim/BQ amXTTQ`i

kX/p M+2/-_HB+2Mb2/ mM/2` i?2 *`2 iBp2 *QKKQMb ii`B#miBQM@ LQM*QKK2`+B H@a? `2 HBF2 9Xy AMi2`M iBQM H Sm#HB+ GB+2Mb2

jX_ 7Q` . i a+BH2M+2Mb2/ *`2 /Bp2 *QKKQMb ii`B#miBQM@
LQM*QKK2`+B H@LQ.2`Bpb jXy

9X JQ/2`M.Bp2 BMiQ _ M/ i?2 iB/vp2`b2

8X9 _ J `F/QrM

h?`Qm;?Qmi i?Bb +Qm`b2- r2 rBHH #2 mbBM; _ J `F/QrM- rBi? ` 2 MQi2b M/ ?QK2@
rQ`F bbB;MK2MibX _ J `F/QrM /Q+mK2Mib +QK#BM2 2t2+mi #H2
M/ T`Qb2 +QKK2Mi `v BMiQ QM2 /Q+mK2MiX h?BMF Q7 M _ J `F/
KQ/2`M / v H # MQi2#QQF- r?2`2 vQm + M + Tim`2 MQi QMHv r? i
HbQ r? i vQm r2`2 i?BMFBM;X

h?2 }H2M K2 Q7 M _ J `F/QrM /Q+mK2Mi b?QmH/ 2M/ BM
+ M HbQ #2 +QMp2`i2/ iQ M QmiTmi 7Q`K i- HBF2 S.6- >hJG- bHB
}H2b- M/ KQ`2X

_ J `F/QrM /Q+mK2Mib +QMi BM i?`22 BKTQ`i Mi ivT2b Q7 +QMi2M

RXM UQTiBQM HV u JG ?2 /2` bm``QmM/2/ #v
kX *?mMFb Q7 _ +Q/2 bm``QmM/2/ #v
jXh2ti KBt2/ rBi? bBKTH2 i2ti 7Q`K iiBM; HBF2 B bn

8X9XR u JG ?2 /2`

u JG bi M/b 7Q` óv2i MQi?2` K `FmT H M;m ;2ôX _ J `F/QrM mb2b B
K Mv /2i BHb Q7 i?2 QmiTmiX

```
@@@
iBiH2, ]>QK2rQ`F R]
mi?Q`, ]_Q+?2HH2 h2`K M]
/ i2, ]6 HH kyRN]
QmiTmi, ?iKHn/Q+mK2Mi
@@@
```

AM i?Bb 2t KTH2- r2 bT2+B}2/ i?2 /Q+mK2Miöb iBiH2- mi?Q`- M
bT2+B}2/ i? i r2 r Mi Bi iQ 2p2Mim HHv +QMp2`i Bi BMiQ M >hJG /

8X9Xk J `F/QrM

S`Qb2 BM _J/ }H2b Bb r`Bii2M BM J `F/QrM- HB;?ir2B;?i b2i Q7 +QM
7Q` 7Q`K iiBM; TH BM i2ti }H2bX J `F/QrM Bb /2bB;M2/ iQ #2 2 b
2 bv iQ r`Bi2X Ai Bb HbQ p2`v 2 bv iQ H2 `MX h?2 ;mB/2 #2HQr b
S M/Q+öb J `F/QrM- bHB;?iHv 2ti2M/2/ p2`bBQM Q7 J `F/QrM i? i
mM/2`bi M/bX

h2ti 7Q`K iiBM;
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Bi HB+ Q` nBi HB+n
#QH/ nn#QH/nn
<+Q/2<
bmT2`b+`BTi k M/ bm#b+`BTi k

>2 /BM;b
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

O Rbi G2p2H >2 /2`

OO kM/ G2p2H >2 /2`

OOO j`/ G2p2H >2 /2`

GBbib
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

"mHH2i2/ HBbi Bi2K R

Ai2K k

Ai2K k

Ai2K k#

RX LmK#2`2/ HBbi Bi2K R

RX Ai2K kX h?2 MmK#2`b `2 BM+`2K2Mi2/ miQK iB+ HHv BM i?2 QmiTm

GBMFb M/ BK ;2b
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

I?iiT,ff2t KTH2X+QK=

(HBMF2/ T?` b2)U?iiT,ff2t KTH2X+QKV

5(QTiBQM H + TiBQM i2ti)UT i?fiQfBK;XTM;V

h #H2b
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

6B`bi >2 /2` % a2+QM/ >2 /2`

```
@@@@@@@@@@@@@ % @@@@@@@@@@@@@
*QMi2Mi *2HH % *QMi2Mi *2HH
*QMi2Mi *2HH % *QMi2Mi *2HH
```

h?2 #2bir viQ H2 `Mi?2b2 Bb bBKTHv iQ i`v i?2K QmiX AirBHHi F2
bQQM i?2v rBHH #2+QK2 b2+QM/ M im`2- M/ vQm rQMöi M22/ iQ i?
A7 vQm 7Q`;2i- vQm + M ;2i iQ ? M/v `272`2M+2 b?22i rBi? >2HT
ZmB+F _272`2M+2X

8X9Xj *Q/2 *?mMFb

hQ `mM +Q/2 BMbB/2 M _ J `F/QrM /Q+mK2Mi- vQm /Q BiBMbB/2 ó
Q7 +?mMF HBF2 bi2T BM H `;2` T`Q+2bbX +?mMF b?QmH/ #
+QMi BM2/- M/ 7Q+mb2/ `QmM/ bBM;H2 i bFX

*?mMFb #2;BM rBi? ?2 /2` r?B+? &+QMbBbib/Q7 M QTiBQM H
+?mMF M K2- 7QHHQr2/ #v +QKK b2T ` i2X 1 +? +QMb2b 7QHHQr2/ #
vQm` _ +Q/2 M/ i?2 +?mMF 2M/ Bb BM/B+ i2/ #v }M H

uQm + M +QMiBMm2 iQ `mM i?2 +Q/2 mbBM; i?2 F2v#Q `/ b?Q`i+mi
2 `HB2`K/f*i`H Y 1MiQm + M HbQ `mM i?2 2MiB`2 +?mMF #v +HB+
_mM B+QM UBi HQQFb HBF2 TH v #miiQM i i?2 iQT Q7 i?2 +?mM
*K/f*i`H Y a?B7i YX1Mi2`

_aim/BQ 2t2+mi2b i?2 +Q/2 M/ /BbTH vb i?2 `2bmHib BMHBM2 rBi

8X9X9 EMBiiBM;

hQ T`Q/m+2 +QKTH2i2 `2TQ`i +QMi BMBM; HH i2ti- +Q/2- M/ `2bmHib-
óEMBiô #miiQM i i?2 iQT Q7 i?2 b+`BTi 2/BiQ` UHQQFb HBF2 # HH Q7 v
*K/f*i`H Y a?B7i Y?Bb rBHH /BbTH v i?2 `2TQ`i BM i?2 pB2r2` T M2- M/
+`2 i2 b2H7@+QMi BM2/ >hJG }H2 i? i vQm + M T`BMi- 2KBH `2 rBi? Qi?2`bX h?2
}H2 Bb r`Bii2M BM i?2 b K2 /B`2+iQ`v b vQm`

8X9X8 *?2 ib?22ib M/ Pi?2` _2bQm`+2b

q?2M rQ`FBM; BM _aim/BQ- vQm + M }M/ M _J `F/QrM +?2 ib?22i #v ;QBM
>2HT = *?2 ib?22ib = _J `F/QrM *?2 i a?22iX

?2HT7mH Qp2`pB2r Q7 _J `F/QrM +_ 7Q`K iiBM; M/ Qmi Tmi

/22T /Bp2 BMiQ _J `F/QrM +?2 #2 7QmM/

8X9Xe *? HH2M;2b

*? HH2M;2 RX

*`2 i2 M2r _J `F/QrM /Q+mK2Mi rBi? 6BH2 = _J `F/QrM
_2 / i?2 BMbi`m+iBQMbX S` +iB+2 `mMMBM; i?2 +?mMFbX

LQr //bQK2 M2r K `F/QrMX h`v //BM; bQK2 }`bi@- b2+QM/@- M/ i?B`/@H
?2 /2`bX AMb2`i HBMF iQ r2#bBi2X

*? HH2M;2 kX

AM i?2 }`bi +Q/2 +?mMF- KQ/B7v _2@`mM i?2 +?mMF- M/ b22
KQ/B}2/ QmiTmiX

*? HH2M;2 jX

EMBi i?2 /Q+mK2Mi BMiQ M S.6 }H2X o2`B7v i? i vQm + M KQ/B7v i?2 BMT
b22 i?2 QmiTmi mT/ i2X

+FMQrH2/;K2Mib

h?Bb T ;2 Bb BM T `i /2`Bp2/ 7`QK i?2 7QHHQrBM; bQm`+2b,

RX 7Q` . i a+B2M+2 Mb2/ *`2 iBp2 *QKKQMb ii`B#miBQM@
LQM*QKK2`+B H@LQ.2`Bpb jXy

*? Ti2` e

_ avMi t

6`mbi` iBQM Bb M im` H r?2M vQm bi `i T`Q;` KKBM; BM _X _ Bb biB+F
TmM+im iBQM- M/ 2p2M QM2 +? ` +i2` Qmi Q7 TH +2 rBHH + mb2 Bi iQ +Q
r?BH2 vQm b?QmH/ 2tT2+i iQ #2 HBiiH2 7`mbi` i2/-i F2 +QK7Q`i BM i? i
ivTB+ H M/i2KTQ` `v, Bi ? TT2Mb iQ 2p2`vQM2- M/i?2 QMHv r v iQ ;2i Q
Bb iQ F22T i`vBM;X

hQ mM/2`bi M/ +QKTmi iBQMb BM _- irQ bHQ; Mb `2 ??2HT7mH, @ 1p
2`vi?BM; i? i 2tBbib Bb M Q#D2+iX @ 1p2`vi?BM; i? i ? TT2Mb Bb
7mM+iBQM + HHX

CQ?M *? K#2`b

eXR o `B #H2b

eXRXR `Bi?K2iB+

AM Bib KQbi # bB+ 7Q`K- _ + M #2 mb2/ b bBKTH2 + H+mH iQ`X *QMb
7QHHQrBM; `Bi?K2iB+ QT2` iQ`b,

- Ç //BiBQYM,
- Ç am#i` +iB@QM,
- Ç JmHiBTHB+ iBQM,
- Ç .BpBbBBQM,
- Ç 1tTQM2MiB iBQM,
- Ç JQ/mHQ U`2K WBW/2`V,

```
Rf kyy jy
O= (R) yXR8
UBNY dj Y kVf j
```

```
O= (R) 99Xd
8 WWk
O= (R) R
```

"mir?2M r2/Q i?Bb- MQM2 Q7 Qm``2bmHib `2b p2/7Q` H i2`mb2

eXRXk bbB;MBM; o`B #H2b

M 2bb2MiB H T `iQ7 T`Q;` KKBM; Bb +`2 iBM; Q# D2+ib UQ` p `B `2M K2b 7Q` p Hm2bX

p `B #H2 Bb +`2 i2/r?2M p Hm2 Bb bbB;M2/iQ BiX q2 /Q i? i rB

```
t l@j
```

l@Bb i?2 + HH2/ bbB;MK2Mi QT2` iQ`X Mb p Hm2b QM i?2 `B;?i iQ Q#D2+ QM i?2 H27i- HBF2 i?Bb,

```
Q#D2+inM K2 l@ p Hm2
```

aQ- 7i2` 2t2+miBM; i?2 p Hm2 Q7h?2 ``Qr + M #2 `2 / b j;Q2b BM tX

LQi2,QMöi4 bbB;MK2MibX AirBHH rQ`F BM bQK2 +QMi2tib- #m +QM7mbBQM H i2`X

q2 + M mb2 p `B #H2b BM + H+mH iBQMb Dmbi b B7 i?2v r2`2 p Hm

```
t l@j
t Y 8
O= (R) 3
```

AMbT2+i Q#D2+ib iQ /BbTH v p Hm2bX

AM _- i?2 +QMi2Mib Q7 M Q#D2+i + M #2 T`BMi2/ #v bBKTHv 2t2+ M K2X

```
t l@j
t
O= (R) j
```

q?Bi2bT +2 K F2b +Q/2 2 bB2` iQ `2 /X

LQiB+2 i? i r2 bm``QmM/2/ i?2 bbB;MK2Mi QT2` iQ` +2 bT +2 Bb B;MQ`2/ UmMHBF2 Svi?QMVX Ai Bb ;QQ/ T` +iB+2 iQ mb2 bT +2b- #2+ mb2 Bi K F2b +

[R] h2+?MB+ HHv- Q#D2+ib M/ p `B #H2b `2 /Bz2`2Mi i?BM;b- #mi r2öHH mb2 i?2 irQ 7Q` MQrX

```
2tT2`BK2Mi@ ]+m`2Mi pbX pQHi ;2]Bb Bb # /
2tT2`BK2Mi@ ]+m`2Mi pbX pQHi ;2]Bb Bb #2ii2`
```

eXRXj o `B #H2 L K2b

P#D2+iM K2b + M QMHv +QMi BMMHX2Xi2`b- MmK#2`b-

uQm r Mi vQm` Q#D2+iM K2b iQ #2 K2 MBM;7mH- `2Bi`B/pp2XB #H2 M K2
UbQ``v5VX uQmöHH HbQ M22/ +QMp2MiBQM 7Q` KmHiBTH2 rQ`/bX
bM F2n+ b2 vQm b2T ` i2 HQr2`+ b2 rQ`/b rBi?

```
BnmbS2nbM F2n+ b2
Qi?2`S2QTH2lb2* K2H* b2
bQK2XT2QTH2Xmb2XT2`BQ/b
M/n 62rXS2QTH2n_1LPlL*1+QMp2MiBQM
```

G2iöb K F2 M bbB;MK2Mi mbBM; bM F2n+ b2,

```
`n`Q+Fb kl@j
```

M/ H2iöb i`v iQ BMbT2+i Bi,

```
`n`Q+F
O= 1``Q` BM 2p HU2tT`- 2MpB`- 2M+HQbV, Q#D2+i ^`n`Q+F^ MQi 7QmM
_n`Q+Fb
O= 1``Q` BM 2p HU2tT`- 2MpB`- 2M+HQbV, Q#D2+i ^_n`Q+Fb^ MQi 7Qm
```

_ Bb + b2@b2MbBiBp25

lb2 i?2 h " F2v iQ miQ+QKTH2i2X

"2+ mb2 ivTQb `2 i?2 #bQHmi2 rQ`bi- r2 + M mb2 _ aim/BQ iQ ?2HT mb ivT2
BMbT2+iBM; _aim/BQöb i # +QKTH2iBQM 7 +BHBivX hvT2 ó`nô- T`2bb
// +? ` +i2`b mMiBH vQm ? p2 mMB[m2 T`2}t- i?2M T`2bb `2im`MX

```
`n`Q+Fb
O= (R) 3
```

eXRX9 *? HH2M;2b

*? HH2M;2 R, J FBM; M/ S`BMiBM; o `B #H2bX

J F2 j p `B #H2b, M K2 UrBi? vQm` 7mHH M K2V- +Biv Ur?2`2 vQm r2`2 #Q
v2 ` Ur?2M vQm r2`2 #Q`MXV

*? HH2M;2 k, ar TTBM; o Hm2b

.`r i #H2 b?QrBM; i?2 p Hm2b Q7 i?2 p `B #H2b BM i?Bb T`Q;` bi i2K2Mi Bb 2t2+mi2/X

AM bBKTH2 i2`Kb- r? i /Q i?2 H bi i?`22 HBM2b Q7 i?Bb T`Q;` K /Q

```
HQr2bi I@ Ry
?B;?2bi I@ jXy
i2KT I@ HQr2bi
HQr2bi I@ ?B;?2bi
?B;?2bi I@ i2KT
```

*? HH2M;2 j, S`2/B+iBM; o Hm2b

q? i Bb i?2 }M H p Hm2 Q7 TQbBiBQM i?2 T`Q;` K #2HQr\ h`v iQ T`2/B+i i?2 p Hm2 rBi?Qmi `mMMBM; i?2 T`Q;` K- i?2M +?2+F vQm` T`2/B+iBQ

```
BMBiB H I@ ]H@7i]
TQbBiBQM I@ BMBiB H
BMBiB H I@ ]`B;?i]
```

*? HH2M;2 9, avMi t

q?v /Q2b i?2 7QHHQrBM; +Q/2 7 BH\

```
;2 44jR
```

M/ i?2 7QHHQrBM;\

```
jR I@ ;2
```

eXk 6mM+iBQMb

_?b H `;2 +QHH2+iBQM Q7 #mBHi@BM 7mM+iBQMb i? i ?2HTb m mb2 7mM+iBQM- r2 HBbi Bib M K2 7QHHQr2/ #v T `2MMi?2b2b +QMi BMBM; Bib `;mK2Mi @ 7mM+iBQMX

```
7mM+iBQMU `;mK2Mi4p HR- `;k 4p Hk- XXXV
```

>2`2 `2 bQK2 ?2HT7mH #mBHi@BM 7mM+iBQMb,

```
Kvnp ` I@ +UR- 8- k- 9- 8V

bmKUKvnp `V
O= (R) Rd
H2M;i?UKvnp `V
```

```
O= (R) 8
KBUMUKvnp `V
O= (R) R
K tUKvnp `V
O= (R) 8
mMB[UmK2vnp `V
O= (R) R 8 k 9
```

eXkXR `;mK2Mib

M `;mK2Mib p Hm2 iT?bibBbMiQ 7mM+iBQMX 1p2`v 7mM+iBQM `2im`Mb `2bmHiX

G2iöb i`v mbBM; b2[UMvT?B+? K F2b `2;mH ` b2[m2M+2b Q7 MmK#2`bX q?BH2 r2ö`2 i Bi- r2öHH H2 `M KQ`2 ??HT7mH 72 im`2b Q7 _aim/BQX

hvT2 b2 M/ ?Bi h "X TQTmT b?Qrb vQm TQbbB#H2 +QKTH2iBQMbX aT2+B7v #v ivTBM; KQ`2 U ó[ôV iQ /Bb K#B;m i2- Q` #v mbBM; ºf· ``Qrb iQ b2H2+iX i?2 ~Q iBM; iQQHiBT i? i TQTb mT- `2KBM/BM; vQm Q7 i?2 7mM+iBQMöb `; Tm`TQb2X

S`2bb h " QM+2 KQ`2 r?2M vQmöp2 b2H2+i2/ i?2 7mM+iBQM vQm r MiX _ rBHH // K i+?BM; U/VT2MUB+MUbVBTM;`2UMi?2b2b 7Q` vQmX hvT2 i?2 `;mK2MiRi-bRyM/ ?Bi `2im`MX

```
b2[UR- RyV
O= (R) R k j 9 8 e d 3 N Ry
```

>Qr K Mv `;mK2Mib /B/ r2 Tbb2[7mM+iBQM\

eXkXk aiQ`2 6mM+iBQM PmiTmi

LQiB+2- r?2M r2 +bb2[7mM+iBQM- MQi?BM; +? M;2/ BM Qm` 2MpB`QMK2MiX h? iöb #2+ mb2 r2 /B/Möi b p2 Qm` `2bmHib iQ M Q#D2+iX G2iöb i`v Bi bbB;MBM; p `B #H2X

```
v l@b2[UR- RyV
v
O= (R) R k j 9 8 e d 3 N Ry
```

eXkXj `;mK2Mi _2bi`B+iBQMb M/ .27 mHib

G2iöb mb2 MQi?2` 7mM+iBQM- + HH2/

```
`QmMU+y XRVkj
O= (R) ey
```

`QmMKmbi#2 ;Bp2M iH2 biQM2 `;mK2MiX M/ BiKmbi#2 ;Bp2M i
+ M #2 K2 MBM;7mH `QmM/2/X

```
`QmMUMV
`QmMU/^V
```

6mM+iBQMb K2 ?mi HH2 7Q` bQK2 `;mK2MibX

"v /27 mHi- `QmM/ Mv MmK#2` iQ x2`Q /2+BK H TH +2bX
+ M bT2+B7v i?2 MmK#2` Q7 /2+BK H TH +2b r2 r MiX

```
`QmMU+y XRVkj
O= (R) ey
`QmMU+y XR-k/jB;Bib kV
O= (R) eyXR
`QmMU+y XR-kjV
O= (R) eyXR
```

eXkX9 .Q+mK2Mi iBQM M/ >2HT 6BH2b

>Qr /Q r2 FMQr r? i FBM/b Q7 `;mK2Mib iQ T bb BMiQ 7mM+iBQM\
7mM+iBQM +QK2b rBi? /Q+mK2Mi iBQMX

Ç\ Y Q#D2+i QT2Mb ?2HT T ;2 7Q` i? i bT2+B}+ Q#D2+i
Ç\\ Y Q#D2+i b2 `+?2b ?2HT T ;2b +QMi BMBM; i?2 M K2 Q7 i?2

```
\K2 M
\\K2 M
```

HH ?2HT }H2b `2 bi`m+im`2/ i?2 b K2 ;mK2Mib- H2 ;mK2Mib- BM7Q`K iBQM
2t +iHv r? i FBM/ Q7 BM7Q`K iBQM r2 + M T bb BMiQ 7mM+iBQ
b2+iBQM 2tTH BMb r? i i?2 QmiTmi Q7 i?2 7mM+iBQM BbX h?2
`2 H 2t KTH2b Q7 i?2 7mM+iBQM BM mb2X

eXkX8 *? HH2M;2b

*? HH2M;2 R, q? i > TT2Mb q?2M

1tTH BM- BM bBKTH2 i2`Kb- i?2 Q`/2` Q7 QT2` iBQMb BM i?2 7Q
r?2M /Q2b i?2 //BiBQM ? TT2M- r?2M /Q2b i?2 bm#i` +iBQM ? TT
2 +? 7mM+iBQM + HH2/- 2i+X

q? i Bb i?2 }M H p Hm2 Q7

```
` /B M+2 R@y
` /B M+2 K@UkX-RkXyY KBU` /B M+2XR ` /B M+@yX8V
```

*? HH2M;2 k, q?v\

_mM i?2 7QHHQrBM; +Q/2X

```
`B+? l@]QH/]
TQQ` l@]iBM]
K tU`B+?- TQQ`V
```

lbBM; i?2 ?2HTK-2bt7Q`BM r?v Bi `2im`Mb i?2 `2bmHi Bi /Q2bX

eXj .i hvT2b

1p2`v pHm2 BM T`Q;` K ?vbT2 bATM+B}+?Qb2 ivT2b `2 + HH2/ ó+H bb2bô- M/ i?2`2 `2 9Q7 i?2K,

- Ç +? ` +i2` Ui2ti Q` óbi`BM;ôV
- Ç MmK2`B+ UBMi2;2` Q` /2+BK HV
- Ç BMi2;2` UDmbi BMi2;2`V
- Ç HQ;B+ H Uh_l1 Q` 6 Ga1 #QQH2 MbV

1t KTH2	hvT2
ó ô- óbr+ô	+? ` +i2`
k- R8X8	MmK2`B+
k UJmbi /G i2M/ iQ /2MQi2 BMi2;2`V	BMi2;2`
h_l16 Ga1	HQ;B+ H

eXjXR q? iöb i? i hvT2\

_ Bb /vM KB+ HHv ivT2/- K2 MBM; i? i Bi ó;m2bb2bô r? i +H bb p Hm2 Bb

lb2 i?2 #mBHi@BM 7mM+iBQM +H bb UV iQ }M/ Qmi r? i ivT2 p Hm2 ? bX

```
+H bbUV
O= (R) ]MmK2`B+]
+H bbUjGV
O= (R) ]BMi2;2`]
+H bbU]h?`2V]
O= (R) ]+? ` +i2`]
+H bbUhV
O= (R) ]HQ;B+ H]
```

h?Bb rQ`Fb QM p `B #H2b b r2HH Xp 'Hm2b K2K#2`, i??22
p `B #H2b Dmbi H #2HX

```
i?`22 l@j
+H bb Ui?`22V
O= (R) ]MmK2`B+]

i?`22 l@]i?`22]
+H bb Ui?`22V
O= (R) ]+? ` +i2`]
```

p Hm2öb +H bb /2i2`KBM2b r? i i?2 T`Q;` K + M /Q iQ BiX

```
j @R
O= (R) k
j @]R]
O= 1``Q` BM j @ ]R], MQM@MmK2`B+ `;mK2Mi iQ #BM `v QT2`
```

eXjXk *Q2`+BQM

q2 Dmbi H2 `M2/ r2 + MMQi bm#i` +i MmK#2`b bX Xbi`BM;bX AMbi2
M K2 Q7 +H bb b 7mM+iBQMb iQ +QMp2`i p Hm2 iQ i? i ivT2X

```
j @ bXMmK2`B+U]R]V
```

h?Bb Bb +H2 `2/ +Q2`+BQM 2öb MQi?2` 2t KTH2,

```
Kvnp ` l@ ]6 Ga1]
Kvnp `
O= (R) ]6 Ga1]
bXHQ;B+ UKvnp `V
O= (R) 6 Ga1
```

q? i /Bz2`2M+2 /B/ vQm MQiB+2\

eXjXj Pi?2` P#D2+ib

h?2`2 `2 72r Qi?2` óQ// # HHô ivT2b BM _,

L `2 KBbbBM; p Hm2b

JBbbBM; p Hm2b `2 bT2+B}2/ rBi? L r?B+? vb #2 +Q2`+2/ iQ i?2 +Q``2+i ivT2 B7 mb2/ BMbB/2

```
t l@+UL- RV
t
O= (R) L R
```

```
ivT2QL7V
O= (R) ]HQ;B+ H]
ivT2QU7V
O= (R) ]/Qm#H2]
```

AMB7b BM}MBivX

uQm + M ? p2 2Bi?2` TQbBiBp2 Q` M2; iBp2 BM}MBivX

```
Rfy
O= (R) AM7
RfAM7
O= (R) y
```

L LK2 Mb óLQi MmK#2`ôX Aiöb M mM/2}M2/ p Hm2X

```
yfy
O= (R) L L
```

eXjX9 *? HH2M;2b

*? HH2M;2 R, J FBM; M/ *Q2`+BM; o `B #H2b

RX J F2 p `B #H2 M/ bbB;M Bi b i?2 v2 ` vQm r2`2 #Q`MX
kX *Q2`+2 i? i p `B #H2 iQ bi`BM;- M/ bbB;M Bi iQ M2r p `B #H2
  v2 `nbi`BM;
jXa QK2QM2 BM vQm` +H bb b vb i?2v r2`2 #Q`M BM kyyRX _2 HHv\ _2 HI
  Qmi r? i vQm` ;2 /Bz2`2M+2 v2 `b-nbi`BM; QMHv

*? HH2M;2 k, 6Bt i?2 *Q/2

*? M;2 i?2 7QHHQrBM; +Q/2 iQ hKlX 2 i?2 QmiTmi

```
p HRnI@6
p HknI@]6]

+H bbUp HRV 44+H bbUp HkV
O= (R) 6 Ga1
```

*? Ti2` d

. i *H bb2b M/ ai`m+im`2b

hQ K F2 i?2 #2bi mb2 Q7 i?2 _ H M;m ;2- vQmöHH M22/ bi`QM; mM/2`bi M
bB+ / i bi`m+im`2b- M/ ?Qr iQ QT2` i2 QM i?2KX

h?Bb+?BbiB+QHH M/2`bi M/ #2+ mb2 i?2b2 `2 i?2 Q#D2+ib vQm rBHH K MBTm
QM / v@iQ@/ v # bBb BM _X "mi i?2v `2 MQi Hr vb b 2 bv iQ rQ`F rBi?
i?2v b22K i i?2 Qmib2iX .2 HBM; rBi? Q#D2+i ivT2b M/ +QMp2`bBQMb B
i?2 KQbi +QKKQM bQm`+2b Q7 7`mbi` iBQM 7Q` #2;BMM2`bX

_öb # b2 / i bi`m+im`2b + M #2 Q`; MBb2/ #v i?2B` /BK2MbBQM HBiv UR/
M/V M/ r?2i?2` i?2vö`2 ?QKQ;2M2Qmb U HH +QMi2Mib Kmbi #2 Q7 i?2 b K
Q` ?2i2`Q;2M2Qmb Ui?2 +QMi2Mib + M #2 Q7 /Bz2`2Mi ivT2bVX h?Bb ;Bp2
}p2 / i ivT2b KQbi Q7i2M mb2/ BM / i M HvbBb,

	>QKQ;2M2Qmb	>2i2`Q;2M2Qmb
R/	iQKB+ p2+iQ`	GBbi
k/	J i`Bt	. i 7` K2
M/	`` v	

1 +? / i bi`m+im`2 ? b Bib QrM bT2+B}+ iBQMb M/ #2? pBQ`X AM i?2 `
i?Bb +? Ti2`- r2 rBHH +Qp2` i?2 ivT2b Q7 / i Q#D2+ib i? i 2tBbi BM _ M/
ii`B#mi2bX

RX b2+iQ`b
kX GBbib
jX6 +iQ`b
9XJ i`B+2b
8X i 7` K2b

dXR o2+iQ`b

G2iöb bi `irBi? QM2@/BK2MbBQM H UR/V Q#D2+ibX h?2`2 `2 irQ

RXiQKB+ p2+iQ`b p2+iQ`b HbQ + HH2/ bi QKB+ HBbib-
kXGBbib- GBbib `2 /BbiBM+i 7`QK iQKB+ p2+iQ`b #2+ mb2 HBbib
Qi?2` HBbibX

q2öHH /Bb+mbb iQKB+ p2+iQ`b }`bibX

dXRXR *`2 iBM; o2+iQ`b

o2+iQ`b `2 R@/BK2MbBQM H +? BMb Q7 p2+iQ`b HH2 +? p
p2+iQ`X

iQKB+ p2+iQ`b `2 mbm HHv +`2 i2/ rBi? Q`i 7Q` õ+QK#BM2ö,

```
t l@+UR- k- jV
t
O= (R) R k j
H2M;i?UtV
O= (R) j
```

q2 + M HbQ // 2H2K2Mib iQ i?2 2M/ Q7 p2+iQ` #v T bbBM; i?2 Q`
BMiQ +?2 M+iBQM- HBF2 bQ,

```
x l@+U]"2vQM+2] ]E2HHv] ]JB+?2HH2] ]G2hQv ]V
x l@+Ux- ]6`` ?]V
x
O= (R) ]"2vQM+2] ]E2HHv] ]JB+?2HH2] ]G2hQv ] ]6`` ?]
```

LQiB+2 i? i p2+iQ`b `2 Hrvb ~ i+ 2M2/ B7 vQm M2bi

```
O i?2b2 `2 2[mBp H2Mi
+UR- +Uk- +Uj- 9VVV
O= (R) R k j 9
+UR- k- j- 9V
O= (R) R k j 9
```

dXRXk L KBM; o2+iQ`

q2 + M HbQ ii +? M K2b iQ Qm` p2+iQ`X h?Bb ?2HTb mb mM/2`bi
2H2K2Mi `272`b iQX

uQm + M ;Bp2 M K2 iQ i?2 2H2K2Mib Q7 p2+iQ` rBi? i?2
> p2 HQQF i i?Bb 2t KTH2,

```
/ vbnKQMi?U I@ k3 jR jy jR jy jR jR jy jR jy jR V
M KUb/ vbnKQMi?U I@ ]]62# ]]J `] ] T`] ]J v] ]CmM ]CmH ] m; ] ]a2T ]]P+i ]]LQ

/ vbnKQMi?
O= C M 62# J ` T` J v CmM CmH m; a2T P+i LQp .2+
O= jR k3 jR jy jR jy jR jR jy jR jy jR
```

uQm + M M K2 p2+iQ` r?2M vQm +`2 i2 Bi,

```
bQK2np2+iQ`U I@ 2 ]_Q+?2HH2 h2`K M] ]S`Q72bbBQM Q`/BM B`2]
bQK2np2+iQ`
O= M K2 T`Q72bbBQM
O= ]_Q+?2HH2 h2`K M] ]S`Q72bbQ` 1ti` Q`/BM B`2]
```

LQiB+2 i? i BM i?2 }`bi + b2- r2 bm``QmM/2/ 2 +? M K2 rBi? [mQi iBQM K
"mi r2 /QMöi ? p2 iQ /Q i?Bb r?2M +`2 iBM; M K2/ p2+iQ`X

L K2b /QMöi ? p2 iQ #2 mMB[m2- M/ MQi HH p Hm2b M22/ iQ ? p2 M K2
+B i2/X >Qr2p2`- M K2b `2 KQbi mb27mH 7Q` bm#b2iiBM;- /2b+`B#2/ BM
+? Ti2`X q?2M bm#b2iiBM;- Bi Bb KQbi mb27mH r?2M i?2 M K2b `2 mMB[m

dXRXj * H+mH iBQMb QM o2+iQ`b

PM2 Q7 i?2 KQbi TQr2`7mH i?BM;b #Qmi p2+iQ`b Bb i? i r2 + M T2`7Q`K `
+ H+mH iBQMb QM i?2KX

6Q` 2t KTH2- r2 + M bmK mT HH i?2 p Hm2b BM mK2`B+ H p2+iQ` mbBM

```
I@ U R @ k j V
bmUK V
O= (R) k
```

q2 + M HbQ bmK irQ p2+iQ`bX Ai Bb BKTQ`i Mi iQ FMQr i? i B7 vQm
p2+iQ`b BM _- Bi i F2b i?2 2H2K2Mi@rBb2 bmKX 6Q` 2t KTH2- i?2 7QHHQ
bi i2K2Mib `2 +QKTH2i2Hv 2[mBp H2Mi,

```
+UR- k- jV Y +U9- 8- eV
+UR Y 9- k Y 8- j Y eV
+U8- d- NV
```

dXRX9 hvT2b Q7 o2+iQ`b

aQ i?2`2 `2 i?2`2 `2 7Qm` +QKKQM ivT2b Q7 p2+iQ`b- /2T2M/BM; QM i?2
HQ;B+ H- BMi2;2`- MmK2`B+ H- +? ` +i2`X

ek *> Sh1_dX .h *G aa1a L.ah_l*hl_1a

GQ;B+ H o2+iQ`b

GQ;B+ H p2+iQ`b i F2 QM QM2 Q7 i?`22 TQbbB#H2 p Hm2b,

RX h_l1
kX 6 Ga1
jXL UKBbbBM; p Hm2V

```
+Uh_l1- h_l1- 6 Ga1- LV
O= (R) h_l1 h_l1 6 Ga1 L
```

LmK2`B+ o2+iQ`b

LmK2`B+ p2+iQ`b +QMi BM MmK#2`bX h?2v +M #2 BMi2;2` Q` /Qm#H2 T`2+BbBQM
#2`bV/Qm#H2 MmK#2`b rBi? /2+BK H TQBMibVX AM T` +iB+2- vQm `
+QM+2`M vQm`b2H7 rBi? i?Bb /Bz2`2M+2- #mi Dmbi FMQr i? i i?2v
`2H i2/ i?BM;bX

```
+UR- k- jj8V
O= (R) R k jj8
+U9Xk- 9Xy- eXy- 8jXkV
O= (R) 9Xk 9Xy eXy 8jXk
```

*? ` +i2` o2+iQ`b

*? ` +i2` p2+iQ`b +QMi BM +? ` +i2` UQ` õbi`BM;öV p Hm2bX LQi
? biQ #2 bm``QmM/2/ #v [mQi2 7Q`2 +K KFbX

```
+U]"2vQM+2]- ]E2HHv]- ]JB+?2HH2]- ]G2hQv]V
O= (R) ]"2vQM+2] ]E2HHv] ]JB+?2HH2] ]G2hQv]
```

dXRX8 *Q2`+BQM

q2 + M +? M;2 Q` +QMp2`i p2+iQ` ivT2 mbBM; bQX`X`XX`X

```
MmKnp `+UR- kX8- 9X8V
+H bbUMmKnp `V
O= (R) ]MmK2`B+]
bX+? ` +i2`UMmKnp `V
O= (R) ]R] ]kX8] ]9X8]
```

_2K2K#2` i? i HH 2H2K2Mib Q7 p2+iQ` Kmbi #2 i?2 b K2 ivT2X a
ii2KTi iQ +QK#BM2 /Bz2`2Mi ivT2b- +2`i BM Q#b ~2tB#H2
ivT2X

6Q` 2t KTH2- +QK#BMBM; +? ` +i2` M/ MBi2;2` vB2H/b +? `

```
+U] ]- RV
O= (R) ] ] ]R]
```

:m2bb r? i i?2 7QHHQrBM; /Q rBi?Qmi `mMMBM; i?2K }`bi,

```
+UR X-d] ]V
+Uh_l-1 kV
+U] ]- h_lV
```

h_l1 44 R M/ 6 Ga1 44 yX

LQiB+2 i? i r?2M HQ;B+ H p2+iQ` Bb +Q2`+2/ iQ 2 MmK2`B+ HXl# 2M BMi2;2` Q` /Qm#H2- +QK2b R6 M/ 6Ga#2 +QK2b yX h?Bb Bb p2`v mb27mH 7Q` +QmMiBM; i?2 MmK#2` Q7 h_l1b- 7Q` BMbi M+2 MUV

```
t l@+U6 Ga1 6 Ga1 h_lV
 bXMmKU2tVB+
O= (R) y y R

O hQi H MmK#2` Q7 h_l1b
bmUKtV
O= (R) R

O S`QTQ`iBQM i? i `2 h_l1
K2 UMtV
O= (R) yXjjj
```

*Q2`+BQM Q7i2M ? TT2Mb miQK iB+ HHvX

h?Bb Bb +BKTHB+Bi +Q2`+BQM- BM i?2 K iB+ H Y-mK2#2` BMXU rBHH +Q2`+2 iQ /Qm#H2 Q` BMi2;2`- M/ +Q2`+2 iQ HQ;B+ H QT2` iBQMb U rBHH +Q2`+2 iQ HQ;B+ HX uQm rBHH mbm HHv ;2i r `MBM; K2bb ;2 B7 i KB;?i HQb2 BM7Q`K iBQMX

```
RI ]k]
O= (R) h_l1
]R]= k
O= (R) 6 Ga1
```

aQK2iBK2b +Q2`+BQMb- 2bT2+B HHv MQMb2MbB+ H QM2b- rQMöi rQ`FX

```
t l@+U] ]- ]#]- ]+]V
 bXMmKU2tVB+
O= q `MBM;, L b BMi`Q/m+2/ #v +Q2`+BQM
O= (R) L L L
```

```
bXHQ;B+tH U+tV
O= (R) L L L
```

dXRXe *? HH2M;2b

*? HH2M;2 R, *`2 i2 M/ 2t KBM2 vQm` p2+iQ`

*`2 i2 +? ` +i2` p2+iQ` 7`mBi HH2/QMi BM 9 Q7 vQm` 7 pQ`Bi2 7`mBi
h?2M 2p Hm i2 Bib bi`m+im`2 mbBM; i?2 +QKK M/b #2HQrX

```
O 6B`bi +`2 i2 vQm` 7`mBi p2+iQ`
O uPl_ *P.1 >1_1

O 1t KBM2 vQm` p2+iQ`
H2M;i?U7`mBiV
+H bbU7`mBiV
bi`U7`mBiV
```

*? HH2M;2 k, *Q2`+BQM

```
O RX *`2 i2 p2+iQ` Q7 b2[m2M+2 Q7 MmK#2`b #2ir22M R iQ R
O kX *Q2`+2 i? i p2+iQ` BMiQ +? ` +i2` p2+iQ`
O jX // i?2 2H2K2Mi ]RR] iQ i?2 2M/ Q7 i?2 p2+iQ`
O 9X *Q2`+2 Bi # +F iQ MmK2`B+ p2+iQ`X
```

*? HH2M;2 j, * H+mH iBQMb QM o2+iQ`b

*`2 i2 p2+iQ` Q7 i?2 MmK#2`b RR@ky- M/ KmHiBTHv Bi #v i?2 G
7`QK *? HH2M;2 kX

dXk GBbib

GBbib `2 /Bz2`2Mi 7`QK p2+iQ`b #2+ mb2 i?2v +M ?QH/ 2H2K2Mib + M #
GBbib `2 bQK2iBK2b + HH2/ `2+m`bBp2 p2+iQ`b- #2+ mb2 HBbi
HBbibX h?Bb K F2b i?2K 7mM/ K2Mi HHv /Bz2`2Mi 7`QK p2+iQ`bX

dXkXR *`2 iBM; GBbib

uQm +QMbi`m+i HBbib #v mbBM; +QV

```
t l@HBbiUR- ] ]- h_l1 +U9- 8- eVV
t
O= ((R))
O= (R) R
O=
O= ((k))
O= (R) ] ]
O=
O= ((j))
O= (R) h_l1
O=
O= ((9))
O= (R) 9 8 e
```

dXkXk L KBM; GBbib

b rBi? p2+iQ`b- r2 + M ii +? M K2b iQ 2 +?2H2K2Mi QM Qm` HBbi,

```
KvnHBbi H@UM K2R 2H2KR-
          M K2k 2H2KkV
```

h?Bb +`2 i2b HBbi rBi? +QKTQM2Mib MK2R M/ K2k, 7mHHv MK2/ +QM, A7 vQm r Mi iQ M K2 vQm` HBbib 7i2` vQm öp2 +`2 i2/ i?2K- vQm + M mb2 i?2 7mM+iBQM b vQm /B/ rBi? p2+iQ`bX h?2 7QHHQrBM; +QKK M/b `2 7mHHv iQ i?2 bbB;MK2Mi #Qp2,

```
KvnHBbi H@Ui2H2KR- 2H2KkV
M K2bUKvnHBbiV @ +U]M K2R]- ]M K2k]V
```

dXkXj GBbi ai`m+im`2

p2`v mb27mH iQQH 7Q` rQ`FBM; rBi? HBbib Bb i?2 bi` 7mM+iBQM- r?B+? mb2b QM `2pB2rBM; i?2 bi`m+im`2 Q7 HBbi- MQi i?2 +QMi2MibX

```
t l@HBbiUi 4 +UR- k- jV-
        # 4 +U]>2HHQ]- ]i?2`2]V-
        + 4 R,RyV
bi`UtV
O= GBbi Q7 j
O=  0 , MmK (R,j) R k j
O=  0 #, +?` (R,k) ]>2HHQ] ]i?2`2]
O=  0 +, BMi (R,Ry) R k j 9 8 e d 3 N Ry
```

HBbi /Q2b MQi T`BMi iQ i?2 +QMbQH2 HBF2 p2+iQ`X AMbi2 /- 2
HBbi bi `ib QM M2r HBM2X

```
tXp2+ +@R-k-jV
tXHBbi HBbiU@R-k-jV
tXp2+
O= (R) R k j
tXHBbi
O= ((R))
O= (R) R
O=
O= ((k))
O= (R) k
O=
O= ((j))
O= (R) j
```

GBbib `2 mb2/ iQ #mBH/ mT K Mv Q7 i?2 KQ`2 +QKTHB+ i2/ / i bi
6Q` 2t KTH2- #Qi?/ i 7` K2b M/ HBM2 ` KQ/2H Q#D2+ib U b T`Q
`2 HBbib,

```
?2 /UKi+ `bV
O= KT; +vH /BbT ?T /` i ri [b2+ pb K ;2 ` + `#
O= J x/ _s9 kRXy e Rey RRy jXNy kXek ReX8 y R 9 9
O= J x/ _s9 q ; kRXy e Rey RRy jXNy kX33 RdXy y R 9
O= . ibmM dRy kkX3 9 Ry3 Nj jX38 kXjk R3Xe R R 9 R
O= >Q`M2i 9 .`Bp2 kRX9 e k83 RRy jXy3 jXkR RNX9 R y
O= >Q`M2i aTQ`i #Qmi R3Xd 3 jey Rd8 jXR8 jX99 RdXy y y
O= o HB Mi R3XR e kk8 Ry8 kXde jX9e kyXk R y j R
BbXHBbiUKi+ `bV
O= (R) h_l1
KQ/ H@KUKT; ri- / i 4Ki+ `bV
BbXHBbiUKQ/V
O= (R) h_l1
```

uQm +QmH/ b v i? i HBbi Bb bQK2 FBM/ bmT2` / i ivT2, vQm + M b
Mv TB2+2 Q7 BM7Q`K iBQM BM Bi5

6Q` i?Bb `2 bQM- HBbib `2 2ti`2K2Hv mb27mH BMbB/2 7mM+iBQ
iQ;2i?2` HQib Q7 /Bz2`2Mi FBM/b Q7 `2bmHib BMiQ bBM;H2 Q#D2
`2im`MX

```
KQ/ H@KUKT; ri- / i 4Ki+ `bV
bi`UKQ/V
O= GBbi Q7 Rk
O= 0 +Q277B+B2Mib , L K2/ MmK (R,k) jdXkN @8Xj9
O= XX@ ii`U - ]M K2b]V4 +?` (R,k) ]UAMi2`+2TiV] ]ri]
```

```
O=  0 `2bB/m Hb    , L K2/ MmK (R,jk) @kXk3 @yXNk @kXyN RXj @yXk
O=   XX@  ii`U - ]M K2b]V4 +?` (R,jk) ]J x/  _s9] ]J x/  _s9 q ;] ]. ibmM
O=  0 2772+ib, L K2/ MmK (R,jk) @RRjXe8 @kNXRRe @RXeeR RXejR yX
O=   XX@  ii`U - ]M K2b]V4 +?` (R,jk) ]UAMi2`+2TiV] ]ri] ]] ]] XXX
O=  0 ` MF    , BMi k
O=  0 7Bii2/Xp Hm2b, L K2/ MmK (R,jk) kjXj kRXN k9XN kyXR R3XN XXX
O=   XX@  ii`U - ]M K2b]V4 +?` (R,jk) ]J x/  _s9] ]J x/  _s9 q ;] ]. ibmM
O=  0  bbB;M  , BMi (R,k) y R
O=  0 [`       ,GBbi Q7 8
O=   XX0 [`   , MmK (R,jk- R,k) @8Xe8d yXRdd yXRdd yXRdd yXRdd XXX
O=   XX XX@  ii`U - ]/BKM K2b]V4GBbi Q7 k
O=   XX XX XX0 , +?` (R,jk) ]J x/  _s9] ]J x/  _s9 q ;] ]. ibmM dRy] ]>Q`
O=   XX XX XX0 , +?` (R,k) ]UAMi2`+2TiV] ]ri]
O=   XX XX@  ii`U - ] bbB;M]V4 BMi (R,k) y R
O=   XX0 [` mt, MmK (R,k) RXR3 RXy8
O=   XX0 TBpQi, BMi (R,k) R k
O=   XX0 iQH  , MmK R2@yd
O=   XX0 ` MF , BMi k
O=   XX@  ii`U - ]+H bb]V4 +?` ][`]
O=  0 /7X`2bB/m H  , BMi jy
O=  0 tH2p2Hb, L K2/ HBbiUV
O=  0 + HH    , H M;m ;2 HKU7Q`KmH  4 KT;   ri- / i  4 Ki+ `bV
O=  0 i2`Kb    ,*H bb2b ^i2`Kb^- ^7Q`KmH ^  H M;m ;2 KT;   ri
O=   XX XX@  ii`U - ]p `B #H2b]V4 H M;m ;2 HBbiUKT;- riV
O=   XX XX@  ii`U - ]7 +iQ`b]V4 BMi (R,k- R) y R
O=   XX XX XX@  ii`U - ]/BKM K2b]V4GBbi Q7 k
O=   XX XX XX XX0 , +?` (R,k) ]KT;] ]ri]
O=   XX XX XX XX0 , +?` ]ri]
O=   XX XX@  ii`U - ]i2`KXH #2Hb]V4 +?` ]ri]
O=   XX XX@  ii`U - ]Q`/2`]V4 BMi R
O=   XX XX@  ii`U - ]BMi2`+2Ti]V4 BMi R
O=   XX XX@  ii`U - ]`2bTQMb2]V4 BMi R
O=   XX XX@  ii`U - ]X1MpB`QMK2Mi]V4I2MpB`QMK2Mi, _n:HQ# H1Mp=
O=   XX XX@  ii`U - ]T`2/p `b]V4 H M;m ;2 HBbiUKT;- riV
O=   XX XX@  ii`U - ]/ i *H bb2b]V4 L K2/ +?` (R,k) ]MmK2`B+] ]MmK2`B
O=   XX XX XX@  ii`U - ]M K2b]V4 +?` (R,k) ]KT;] ]ri]
O=  0 KQ/2H  ,^/ i X7` K2^,   jk Q#bX Q7  k p `B #H2b,
O=   XX0 KT;, MmK (R,jk) kR kR kkX3 kRX9 R3Xd R3XR R9Xj k9X9 kkX3
O=   XX0 ri , MmK (R,jk) kXek kX33 kXjk jXkR jX99 XXX
O=   XX@  ii`U - ]i2`Kb]V4*H bb2b ^i2`Kb^- ^7Q`KmH ^  H M;m ;2 KT;
O=   XX XX XX@  ii`U - ]p `B #H2b]V4 H M;m ;2 HBbiUKT;- riV
O=   XX XX XX@  ii`U - ]7 +iQ`b]V4 BMi (R,k- R) y R
O=   XX XX XX XX@  ii`U - ]/BKM K2b]V4GBbi Q7 k
O=   XX XX XX XX XX0 , +?` (R,k) ]KT;] ]ri]
O=   XX XX XX XX XX0 , +?` ]ri]
```

```
O=   XX XX XX@  ii`U - ]i2`KXH #2Hb]V4 +?` ]ri]
O=   XX XX XX@  ii`U - ]Q`/2`]V4 BMi R
O=   XX XX XX@  ii`U - ]BMi2`+2Ti]V4 BMi R
O=   XX XX XX@  ii`U - ]`2bTQMb2]V4 BMi R
O=   XX XX XX@  ii`U - ]X1MpB`QMK2Mi]V4I2MpB`QMK2Mi, _n:H
O=   XX XX XX@  ii`U - ]T`2/p `b]V4 H M;m ;2 HBbiUKT;- riV
O=   XX XX XX@  ii`U - ]/ i *H bb2b]V4 L K2/ +?` (R,k) ]MmK2`B
O=   XX XX XX XX@  ii`U - ]M K2b]V4 +?` (R,k) ]KT;] ]ri]
O=  @  ii`U - ]+H bb]V4 +?` ]HK]
```

dXkX9 *? HH2M;2b

*? HH2M;2 RX

q? i `2 i?2 7Qm` # bB+ ivT2b Q7 iQKB+ p2+iQ`b\ >Qr /Q2b HBbi
iQKB+ p2+iQ`\

*? HH2M;2 kX

q?v BR 44]R`]m2\ q?v@BR I 6 G`m12\ q?v]Q M2] I7kHb2\

*? HH2M;2 jX

*`2 i2 i?`22 p2+iQ`b M/ +QK#BM2 i?2K BMiQ HBbiX bbB;M i?2K

*? HH2M;2 9X

A7 Bb HBbi- r? i Bb i?(R)+H b #Q7m(iR)\

dXj 6 +iQ`b

6 +iQ`b `2 bT2+B H p2+iQ`b i?2;Q`B+ H p2`Bi #H2b i? i ? p2
}t2/ M/ FMQrM b2i Q7 TQbbB#H2 p Hm2bX h?BMF, .2KQ+` i- _2
T2M/2Mic J H2- 62K H2- Pi?2`c 2i+X

Ai Bb BKTQ`i Mi i? i _ FMQrb r?2i?2` Bi Bb /2 HBM; rBi? +QMiB
+ i2;Q`B+ H p `B #H2- b i?2 bi iBbiB+ H KQ/2Hb vQm rBHH /2p2
i`2 i #Qi? ivT2b /Bz2`2MiHvX

>BbiQ`B+ HHv- 7 +iQ`b r2`2 Km+? 2 bB2` iQ rQ`F rBi? i? M +? ` +i
K Mv Q7 i?2 7mM+iBQMb BM # b2 _ miQK iB+ HHv +QMp2`i +? `
h?Bb K2 Mb i? i 7 +iQ`b Q7i2M TQT mT BM TH +2b r?2`2 i?2vö`2 MQ

dXjXR *`2 iBM; 6 +iQ`b

hQ +`2 i2 7 +iQ`b BM _- vQm 7mb2 i?2 7mM+iBQM Q`X 7Q+iQ`UV- r?B+? iF2b Q`2 BM; i? ivQm ? p2 iQ /Q Bb +`2 i2 p2+iQ` i? i +QMi BMb HH i?2 Q#b2`p iBQMb i? i #2HQM; iQ ;`QmT- HBKBi2/ MmK#2` Q7 + i2;Q`B2bX 6Q` 2t KTH2- i?2 T `ivA. Q7 8 /Bz2`2Mi BM/BpB/m Hb,

T `ivnp2+iQ` U@ +U]_2T]-]]_2T]-]].2K]-]]_2T]-]].2K]V

Ai Bb +H2 ` i? i i?2`2 `2 irQ + i2;Q`B2b- Q` H2p2Hb U_2T- .2KV- BM ?2`22 KM/_2T,

h?2 7mM+iBQM Q`/2`b i?2 H2p2Hb HT? #2iB+ HHvX h?2 b2+QM/ `;mK2Mi Q7 7 +iQ` b 7 +iQ`,

T `ivn7 +iQ` I@ 7 +iQ`UT `ivnp2+iQ`V
T `ivnp2+iQ`
O= (R)]_2T]]_2T]].2K]]_2T]].2K]
T `ivn7 +iQ`
O= (R) _2T _2T .2K _2T .2K
O= G2p2Hb, .2K _2T

dXjXk amKK `BxBM; 6 +iQ`

PM2 Q7 vQm` 7 pQ`Bi2 7mM+iBQMb- bmKK `vUV- r?B+? vQm H2 `M2/ BM T`2pBQmb +? Ti2`- mb2b H2p2Hb Q7 7 +iQ` bmKK `BxBM; #Qi? i?2 +? ` +i2` p2+iQ` M/ i?2 7 +iQ`,

bmKK `vUT `ivnp2+iQ`V
O= G2M;i? *H bb JQ/2
O= 8 +? ` +i2` +? ` +i2`
bmKK `vUT `ivn7 +iQ`V
O= .2K _2T
O= k j

dXjXj *? M;BM; 6 +iQ` G2p2Hb

q?2M vQm +`2 i2 i?2 7 +iQ`- i?2 7 +iQ` H2p2Hb `2 b2i iQ bT2+B}+ p Hm2b ++2bb i?Qb2 p Hm2b mbBM; i?2 H2p2Hb 7mM+iBQMX

H2p2HbUT `ivn7 +iQ`V
O= (R)].2K]]_2T]

Mv p Hm2 MQi BM i?2 b2i Q7 H2p2Hb rBHH #2 +QMp2`i2/ iQ r Mi iQ // M AM/2T2M/2Mi iQ Qm` b KTH2,

T `ivn7 +iQ`(8) I@]AM/]
O= q `MBM; BM <(I@X7 +iQ`<U< iKT <- 8- p Hm2 4]AM/]V, BMp HB/ 7 +

```
O= L ;2M2` i2/
T `ivn7 +iQ`
O= (R) _2T _2T .2K _2T IL =
O= G2p2Hb, .2K _2T
```

q2 }`bi M22/ iQ // óAM/ô iQ Qm` 7 +iQ` H2p2HbX h?Bb rBHH HI
AM/2T2M/2Mib iQ Qm` b KTH2,

```
H2p2HbUT `ivn7 +iQ`V
O= (R) ].2K] ]_2T]
H2p2HbUT `ivn7 +iQ`V+U I@K- ]_2T- ]]AM/]V
T `ivn7 +iQ` 8 (@ ]AM/]
T `ivn7 +iQ`
O= (R) _2T _2T .2K _2T AM/
O= G2p2Hb, .2K _2T AM/
```

dXjX9 6 +iQ`b `2 AMi2;2`b

6 +iQ`b `2 T`2iiv Km+? BMi2;2`b i? i ? p2 H #2Hb QM i?2KX IM/2
`2 HHv MmK#2`b UR- k- j VX

```
bi`UT `ivn7 +iQ`V
O= 6 +iQ` rf j H2p2Hb ].2K]-]_2T]-]AM/], k k R k j
```

h?2v `2 #2ii2` i? M mbBM; bBKTH2 BMi2;2` H #2Hb #2+ mb2 7
/2b+`B#BM;X 6Q/`2tK-TH2-2Tm#HB+ MKQ`2 /2b+`BTiBp2 i? M
Rb Mkb X

>Qr2p2`- 7 +iQ`b `2 LPh +? ` +i2`b55

q?BH2 7 +iQ`b HQQF U M/ Q7i2M #2? p2V HBF2 +? ` +i2` p2+iQ`b
BMi2;2`bX "2 + `27mH r?2M i`2 iBM; i?2K HBF2 bi`BM;bX

```
t I@ +U] ]- ]#]- ]#]- ] ]V
t I@ bX7 +iQUtV
+Ut-]+]V
O= (R) ]R] ]k] ]k] ]R] ]+]
```

6Q` i?Bb `2 bQM- Biöb mbm HHv #2ii2` iQ 2tTHB+BiHv +QMp2`i 7 +iQ` p2+@
iQ`b B7 vQm M22/ bi`BM;@HBF2 #2? pBQm`X

```
t I@ +U] ]- ]#]- ]#]- ] ]V
t I@ bX7 +iQUtV
t I@ bX+? ` +i2`UtV
+Ut-]+^]V
O= (R) ] ] ]#] ]#] ] ] ]+^]
```

dXjX8 *? HH2M;2b

*? HH2M;2 RX

q? i? TT2Mb iQ 7 +iQ` r?2M vQm KQ/B7v Bib H2p2Hb\

```
7R l@ +iQ`UH2ii2`bV
H2p2Hb UH7RV l@UH2p2Hb UH7RVV
7R
O= (R) x v t r p m i b ` [ T Q M K H F D B ? ; 7 2 / + #
O= G2p2Hb, x v t r p m i b ` [ T Q M K H F D B ? ; 7 2 / + #
```

*? HH2M;2 kX

q? i /Q2b i?Bb +Q/2 /Q\ 7k >Qr /Q2b 7k /Bz2` 7`QK 7R\

```
7k l@2pH 7 +iQ`UH2ii2`bVV
7j l@7 +iQ`UH2ii2`b- H2p2Hb 24U H2ii2`bVV
```

dX9 J i`B+2b

J i`B+2b `2 k@/ p2+iQ`bX h? i Bb- i?2v `2 +QHH2+iBQM Q7 2H2K2Mib Q7 i?2 b K2 ivT2 UMmK2`B+- +? ` +i2`- Q` HQ;B+ HV- `` M;2/ BMiQ }t2/ MmK#2` Q7 `Qrb M/ +QHmKMbX

"v /2}MBiBQM- B7 vQm r Mi iQ +QK#BM2 /Bz2`2Mi ivT2b Q7 / i UQM2 +QHmKM Q7 MmK#2`b- MQi?2` +QHmKM Q7 +? ` +i2`bV- vQm r Mi /i 7` K2- MQi K i`BtX

dX9XR *`2 iBM; J i`B+2b

q2 + M +`2 i2 K i`Bt mbBM; i?2 K i`BtUV 7mM+iBQMX AM i?Bb 7mM+iBQM- r2 bbB;M /BK2MbBQMb iQ p2+iQ`- HBF2 i?Bb,

```
K l@K i`BtUR,e- M`Qr 4 j- M+QH 4 jV
K
O= (-R) (-k) (-j)
O= (R-) R j 8
O= (k-) k 9 e
```

LQiB+2 i? i K i`B+2b }HH +QHmKM@rBb2X q2 + M +? M;2 i?Bb mbBM; i?2 `;mK2Mi,

```
K l@K i`BtUR,e- #v`Qr 4 h- M`Qr 4 j- M+QH 4 jV
K
O= (-R) (-k) (-j)
```

```
O= (R-)   R   k   j
O= (k-)   9   8   e
```

MQi?2` r v iQ +`2 i2 K i`B+2b Bb iQ #BM/ +@BM/ UKM/b Q` `Qrb mbBM `#BM/XUV

```
t l@R, j
v l@R yR k
+#BM/Ut- vV
O=    t   v
O= (R-) R  Ry
O= (k-) k  RR
O= (j-) j  Rk
O Q`
`#BM/Ut- vV
O=    (-R) (-k) (-j)
O= t    R    k    j
O= v    Ry    RR    Rk
```

dX9Xk J i`Bt .BK2MbBQMb

lb2/BKUV Q }M/ Qmi ?Qr K Mv `Qrb Q` +QHmKMb `2 BM K i`Bt UQ` / i

```
/BKUKV
O= (R) k j
```

q2 + M i` MbTQb2 K i`Bt UQ` /iUV V` K2V rBi?

```
K l@K i`BUR, e- M`Qr k- M+QH jV4
K
O=    (-R) (-k) (-j)
O= (R-)   R   j   8
O= (k-)   k   9   e
iUKV
O=    (-R) (-k)
O= (R-)   R   k
O= (k-)   j   9
O= (j-)   8   e
```

dX9Xj J i`Bt L K2b

Cmbi HBF2 p2+iQ`b Q` HBbib- r2 + M ;Bp2 K i`B+2b M K2b i? i /2b+ +QHmKMb

```
K I@K i`BUR,e- M`Qrk+ M+QHjV4

`QrM KU2bKV I@+]`QrR-]]`QrkV]
+QHM UK2VbI@+] ]- ]"]- ]*]V

K
O=      " *
O= `QrR R j 8
O= `Qrk k 9 e
```

dX9X9 *? HH2M;2

h F2 HQQF i i?2 p2+iQ` Aöp2 +`2 i2/ #Qmi #Qt Q{+2 b H2b 7Q` i?2 }`bi i
> ``v SQii2` KQpB2b,

```
O "Qt Q77B+2 b H2b UBM KBHHBQMb5V
T?BHQbQT?2`bnbiQM2+UR@jR@e8dVk
+? K#2`nb2+`2ib+U8 9@dkeRXN-ReXN
T`BbQM2`n xF #U8X+@k9NX89dVR

O o2+iQ`b `2;BQM M/ iBiH2b- mb2/ 7Q` M KBM;
`2;BQM+U@]E]- ]la]- ]Pi?2`V]
iBiH2b+U@]S?BHQbQT?2`^b ]a?Q#2]` Q7 a2+`2ib]bQM2` Q7 VF # M]
```

uQm` +? HH2M;2 Bb iQ,

RX*QK#BM2 i?2 }`bi i?`22 p2+iQ`b BMiQ K i`Bt
kX// M K2b 7Q` i?2 K i`BtBöbH2ö0rM/U+QHm2K;M2bQVU
jXlb2`QramKibQU}VM/i?2 iQi H qQ`H/rB/2 "Qt P{+2 b H2b 7Q` 2 +? KQpB2X

dX8 .i 7` K2b

/i 7` K2 Bb p2`v BKTQ`i Mi / i ivT2 BM _X/Ai7b+TQ2iiv Km+? i?2
/i bi`m+im`2 7Q` KQbii #mH `/i M/Biöb HbQ r? i r2 mb2 7Q` bi iBbiB+

G2iöb b v r2ö`2 rQ`FBM; rBi? i?2 7QHHQrBM; bm`p2v/i ,

- õ `2vQm K ``B2/\ö Q` õv2bHMQB+[mH2biBQMb U
- õ>Qr QH/ `2 vQmK\2VU+
- õq? i Bb vQm` QTBMBQM QM h`mKT\ö Q` Qi?2` õQT2M@2M/2/ö [m2
  U+? ` +iV2`
-

K i`Bt rQMöi rQ`F ?2`2 #2+ mb2 i?2 / i b2i +QMi BMb /Bz2`2Mi / i ivT2bX

/i 7`K2Bb k@/BK2MiBQM H/i bi`m+im`2 +QMi BMBM; ?2i2`
ivT2bX 1 +? +QHmKM Bb p `B #H2 Q7 /i b2i- M/ i?2 `Qrb `2 Q#

L", uQm KB;?i ? p2 ?2 `/ Q7 óiB##H2b` p2 mb2/ iB2M i?2
Q7 T +F ;2bX hB##H2b `2 HBF2 /i 7` K2b kXy- ir2 FBM; bQK2
#2? pBQ` Q7 /i 7` K2b iQ K F2 HB72 2 bB2` 7Q` /i MHvbBbX
Dmbi i?BMF Q7 iB##H2b M/ /i 7` K2b b i?2 b K2 i?BM; M/ /
rQ``v #Qmi i?2 /Bz2`2M+2X

dX8XR *`2 iBM; . i 7` K2b

_ +QMi BMb MmK#2` Q7 #mBHi@BM /i b2ib i? i `2 biQ`2/ b /i
2t KTH2-Ki?2 `b/ i b2i +QMi BMb BM7Q`K iBQM QM miQKQ#BH2 /2
T2`7Q`K M+2 7Q` jk miQKQ#BH2b,

```
+H bb UKi+ `bV
O= (R) ]/ i X7` K2]
?2 / UKi+ `bV
O=                KT; +vH /BbT ?T /` i ri [b2+ pb K ;2 ` + `#
O= J x/ _s9    kRXy  e  Rey RRy jXNy kXek ReX8 y R   9   9
O= J x/ _s9 q ;   kRXy  e  Rey RRy jXNy kX33 RdXy y R   9
O= . ibmM dRy     kkX3  9  Ry3  Nj jX38 kXjk R3Xe R R   9   R
O= >Q`M2i 9 .`Bp2  kRX9  e  k83 RRy jXy3 jXkR RNX9 R y
O= >Q`M2i aTQ`i #Qmi R3Xd 3 jey Rd8 jXR8 jX99 RdXy y y
O= o HB Mi         R3XR  e  kk8 Ry8 kXde jX9e kyXk R y  j   R
```

q2 HbQ +`2 i2 /i 7` K2b r?2M r2 BKTQ`i /i +bp`QQmm2?
/i }H2 BMTmiX q2öHH i HF KQ`2 #Qmi BKTQ`iBM; /i H i2` BM i?

q2 + M +`2 i2 /i 7` K2 7`QK b/+i`X+? m2XBM?Bb 7mM+iBQM
i F2b p2+iQ`b b BMTmi,

```
O .27BMBiBQM Q7 p2+iQ`b
M K2 I@ +]J2`+m`v]- ]o2Mmb]- ]1 `i?]- ]J `b]- ]CmTBi2`]- ]a im`M]- ]l` Mmb]- ]L2Tim
ivT2 I@ +]h2``2bi`B H TH M2i]- ]h2``2bi`B H TH M2i]- ]h2``2bi`B H TH M2i]- ]h2``2b
/B K2i2` I@ +UyXj3- yXN9- R- yX8j- RRXkR- NXy9- 9XyR- jX33V
`BM;b I@ +U6 Ga1- 6 Ga1- 6 Ga1- 6 Ga1- h_l1- h_l1- h_l1- h_l1V

TH M2ib /I@ /X7` K2UM K2- ivT2- /B K2i2`- `BM;bV
TH M2ib
O=    M K2          ivT2 /B K2i2` `BM;b
O= R J2`+m`v h2``2bi`B H TH M2i   yXj3k 6 Ga1
O= k  o2Mmb h2``2bi`B H TH M2i   yXN9N 6 Ga1
O= j  1 `i? h2``2bi`B H TH M2i   RXyyy 6 Ga1
O= 9   J `b h2``2bi`B H TH M2i   yX8jk 6 Ga1
O= 8 CmTBi2`   : b ;B Mi  RRXkyN  h_l1
O= e  a im`M    : b ;B Mi   NX99N  h_l1
```

```
O= d l` Mmb  : b ;B Mi  9Xyyd h_l1
O= 3 L2TimM2  : b ;B Mi  jX33j h_l1
```

"2r `2,/ i X7` K2 ob V/27 mHi #2? pBQm` im`Mb bi`BM;b BMiQ 7 +iQ`bX lb2
bi`BM; b6 +iQ`b 4iQ bGmaIT`2bb i?Bb #2? pBQm` b M22/2/,

```
TH M2ib/l@X7` K2 UM K2- ivT2- /B K2i2`- bi`BM;b b6 +iQ`b 4 6V
TH M2ib
O=   M K2        ivT2 /B K2i2` `BM;b
O= R J2`+m`v h2``2bi`B H TH M2i   yXj3k 6 Ga1
O= k  o2Mmb h2``2bi`B H TH M2i   yXN9N 6 Ga1
O= j  1 `i? h2``2bi`B H TH M2i   RXyyy 6 Ga1
O= 9   J `b h2``2bi`B H TH M2i   yX8jk 6 Ga1
O= 8 CmTBi2`  : b ;B Mi  RRXkyN h_l1
O= e  a im`M  : b ;B Mi  NX99N h_l1
O= d l` Mmb  : b ;B Mi  9Xyyd h_l1
O= 3 L2TimM2  : b ;B Mi  jX33j h_l1
```

dX8Xk h?2 ai`m+im`2 Q7 . i 7` K2b

lM/2` i?2 ?QQ/- / i 7` K2 Bb HBbi Q7 2[m H@H2M;i? p2+iQ`bX h?Bb K F
k@/BK2MbBQM H bi`m+im`2- bQ Bi b? `2b T`QT2`iB2b Q7 #Qi? i?2 K i`Bt

```
p2+R l@ R j
p2+k l@ ] ]- ]#]- ]+]V
/7 l@ / i X7` K2 Up2+R- p2+kV

bi` U/7V
O= ^/ i X7` K2^,  j Q#bX Q7  k p `B #H2b,
O=  0 p2+R, BMi  R k j
O=  0 p2+k, 6 +iQ` rf j H2p2Hb ] ]-]#]-]+], R k j
```

h?2 H2M;i?U/7V / i 7` K2 Bb i?2 H2M;i? Q7 i?2 mM/2`HvBM; HBbi M/ bQ Bb i
b K2 M+QHU/7V Ub2 p2b i?2 MmK#2` Q7 `QrbX

```
p2+R l@ R j
p2+k l@ ] ]- ]#]- ]+]V
/7 l@ / i X7` K2 Up2+R- p2+kV

O i?2b2 irQ `2 2[mBp H2Mi @ MmK#2` Q7 +QHmKMb
H2M;i?U/7V
O= (R) k
M+QHU/7V
O= (R) k

O ;2i MmK#2` Q7 `Qrb
```

```
M`QrU/7V
O= (R) j

O ;2i MmK#2` Q7 #Qi? +QHmKMb M/ `Qrb
/BKU/7V
O= (R) j k
```

dX8Xj L KBM;. i 7` K2b

GBF2 K i`B+2b- / i 7+ K2b M ?p2 MU/Q rM K2xb >Qr2p2`- bBM+2 / i 7` K2b `2 `2 HHv HBbib UQ7 p2+iQ`bV- i?2 MmKb M/ i?2b ` 2 i?2 b K2 i?BM;X

```
p2+R I@ R
p2+k I@ +] ]- ]#]- ]+]V
/7 I@ / i X7` K2Up2+R- p2+kV

O i?2b2 irQ `2 2[mBp H2Mi
M K2bU/7V
O= (R) ]p2+R] ]p2+k]
+QHM K2bU/7V
O= (R) ]p2+R] ]p2+k]

O +? M;2 i?2 +QHM K2b
+QHM K2bU/7V I@ +]LmK#2`]- ]*? ` +i2`]V
/7
O=   LmK#2` *? ` +i2`
O= R   R
O= k   k      #
O= j   j      +

M K2bU/7V I@ +]LmK#2`]- ]*? ` +i2`]V
/7
O=   LmK#2` *? ` +i2`
O= R   R
O= k   k      #
O= j   j      +

O +? M;2 i?2 `QrM K2b
`QrM K2bU/7V
O= (R) ]R] ]k] ]j]
`QrM K2bU/7V I@ +]/QMmi]- ]TB+F]- ]`2ix2H]V
/7
O=        LmK#2` *? ` +i2`
O= /QMmi   R
```

```
O= TB+FH2 k    #
O= T`2ix2H j   +
```

dX8X9 *Q2`+BM; . i 7` K2b

*Q2`+2 M Q#D2+i iQ / i b7X/Ki2X7B`i?K2UV

- Ç p2+iQ` rBHH +`2 i2 QM2@+QHmKM / i 7` K2X
- Ç HBbi rBHH +`2 i2 QM2 +QHmKM 7Q` 2 +? 2H2K2Mic BiöB M 2``Q` B7 i HH i?2 b K2 H2M;i?X
- Ç K i`Bt rBHH +`2 i2 / i 7` K2 rBi? i?2 b K2 MmK#2` Q7 +QHmKMb M/ `Qrb b i?2 K i`BtX

dX8X8 *? HH2M;2b

*? HH2M;2 RX

*`2 i2 jtk / i 7` K2 # b2/ QM i/h?2 }`bi +QHmKM b?QmH/ +QMi BM i?2 M K2b Q7 j 7`mBibX h?2 b2+QM/ +QHmKM b?QmH/ +QMi BM i?2 T`B+2 Q7 i

*? HH2M;2 kX

LQr ;Bp2 vQm` / i 7` K2 TT`QT`B i2 +QHmKM M/ `Qr M K2bX

*? HH2M;2 jX

// i?B`/ +QHmKM +HH2/ i2HHb K2 r? i +QHQ` 2 +? 7`mBi BbX

dX8Xe ZmBx

uQm + M +?2+F vQm` MbrX`2b BM

RX >Qr Bb HBbi /Bz2`2Mi 7`QK M p2+iQ`\

kXq? i `2 i?2 7Qm` +QKKQM ivT2b Q7 p2+iQ`b\

jXq? i `2 M K2b\ >Qr /Q vQm ;2i i?2K M/ b2i i?2K\

9X>Qr Bb K i`Bt /Bz2`2Mi 7`QK / i 7` K2\

dX8Xd Mbr2`b

RX h?2 2H2K2Mib Q7 HBbi + M #2 Mv ivT2 U2p2M HBbiVc i?2 2
iQKB+ p2+iQ` `2 HH Q7 i?2 b K2 ivT2X

kX h?2 7Qm` +QKKQM ivT2b Q7 p2+iQ` `2 HQ;B+ H- BMi2;2`- /Qm
+ HH2/ MmK2`B+V- M/ +? ` +i2`X

jX L K2b HHQr vQm iQ ii +?H #2HbiQ p Hm2bX uQm + M ;2i M/ b
M K2b rBi? K2bUtV M K2bUtV I@ +U]t]-]v]- XXXV

9X 1p2`v 2H2K2Mi Q7 K i`Bt Kmbi #2 i?2 b K2 ivT2c BM / i 7`
/Bz2`2Mi +QHmKMb + M ? p2 /Bz2`2Mi ivT2bX

*? Ti2` 3

am#b2iiBM;

q?2M rQ`FBM; rBi? / i - vQmöHH M22/ iQ bm#b2i Q#D2+ib 2 `Hv M/ Q7i2MX
_öb bm#b2iiBM; QT2` iQ`b `2 TQr2`7mH M/ 7 biX J bi2`v Q7 bm#b2iiBM
vQm iQ bm++BM+iHv 2tT`2bb +QKTH2t QT2` iBQMb BM r v i? i 72r Qi?2` H
+ M K i+?X am#b2iiBM; Bb ? `/ iQ H2 `M #2+ mb2 vQm M22/ iQ K bi2` Mm
Q7 BMi2``2H i2/ +QM+2Tib,

Ç h?2 i?`22 bm#b2iiBM; QT2` iQ`b, (-(, (-))X

Ç h?2 7Qm` ivT2b Q7 bm#b2iiBM;X

Ç h?2 BKTQ`i Mi /Bz2`2M+2b BM #2? pBQm` 7Q` /Bz2`2Mi Q#D2+ib U2X;
HBbib- 7 +iQ`b- K i`B+2b- M/ / i 7` K2bVX

Ç h?2 mb2 Q7 bm#b2iiBM; BM +QMDmM+iBQM rBi? bbB;MK2MiX

h?Bb mMBi ?2HTb vQm K bi2` bm#b2iiBM; #v bi `iBM; rBi? i?2 bBKTH2bi
bm#b2iiBM;, bm#b2iiBM; M iQKB+ p2+iQ`b rBi? (X Ai i?2M ;2M2` HBx2b vQm`
FMQrH2/;2- }`bi iQ KQ`2 +QKTH2t i2/ / i ivT2b UHBF2 / i 7` K2b M/ HB
M/ i?2M iQ i?2 Qi?2` bm#b2iiBM; QT2` iQ`b- (( M/ ()X LQi2 i?2M H2 `M ?Qr
bm#b2iiBM; M/ bbB;MK2Mi + M #2 +QK#BM2/ iQ KQ/B7v T `ib Q7 M Q#D2
}M HHv- vQmöHH b22 H `;2 MmK#2` Q7 mb27mH TTHB+ iBQMbX

3XR am#b2iiBM; o2+iQ`b

Aiöb 2 bB2bi iQ H2 `M ?Qr bm#b2iiBM; rQ`Fb 7Q` p2+iQ`b- M/ i?2M ?Qr Bi ;
iQ ?B;?2` /BK2MbBQMb M/ Qi?2` KQ`2 +QKTHB+ i2/ Q#D2+ibX q2öHH bi
i?2 KQbi +QKKQMHv mb2/ QT2` iQ`X

3XRXR am#b2iiBM; hvT2b

G2iöb 2tTHQ`2 i?2 /Bz2`2Mi ivT2b Q7 bm#b2XiBM; rBi? bBKTH2 p

```
t l@+UkX R9X kjXj- 8X9
```

LQi2 i? i i?2 MmK#2` 7i2` i?2 /2+BK H TQBMi ;Bp2b i?2 Q`B;BM H p2+iQ`X

h?2`2 `2 7Qm` i?BM;b vQm + M mb2 iQ bm#b2i p2+iQ`,

RX SQbBiBp2 BMi2;2`b `2im`M 2H2K2Mib i i?2 bT2+B}2/ TQbBiBQ

```
Ut l@+UkX R9X kjXj- 8X9V
O= (R) kXR 9Xk jXj 8X9
t(R)
O= (R) kXR
```

q2 + M HbQ BM/2t KmHiBTH2 p Hm2b #v T bbBM; p2+iQ` Q7 BMi2

```
Ut l@+UkX R9X kjXj- 8X9V
O= (R) kXR 9Xk jXj 8X9
t(+Uj- RV)
O= (R) jXj kXR

O .mTHB+ i2/ BM/B+2b vB2H/ /mTHB+ i2/ p Hm2b
t(+UR- RV)
O= (R) kXR kXR
```

LQi2 i? i vQm2 ?Q mH2 +B MbB/2` ?Q i?Bb iQ rQ`F5

JQ`2 2t KTH2b,

```
O <Q`/2`UtV< ;Bp2b i?2 BM/2t TQbBiBQMb Q7 bK HH2bi iQ H `;2
Ut l@+UkX R9X kjXj- 8X9V
O= (R) kXR 9Xk jXj 8X9
Q`/2`UtV
O= (R) R j k 9

O mb2 i?Bb iQ Q`/2` p Hm2bX
t(Q`/2`UtV)
O= (R) kXR jXj 9Xk 8X9
t(+UR- j- k- 9V)
O= (R) kXR jXj 9Xk 8X9
```

kX L2; iBp2 BMi2;2`b QKBi 2H2K2Mib i i?2 bT2+B}2/ TQbBiBQMb,

```
t l@+UkX-R9X-kjXj- 8XV9
t(@R)
O= (R) 9Xk jXj 8X9
t(@+Uj- RV)
O= (R) 9Xk 8X9
```

uQm + Möi KBt TQbBiBp2 M/ M2; iBp2 BMi2;2`b BM bBM;H2 bm#b2i,

```
t l@+UkX-R9X-kjXj- 8XV9
t(+U@R- kV)
O= 1``Q` BM t(+U@R- kV), QMHv y^b K v #2 KBt2/ rBi? M2; iBp2 bm#b+
```

jX *? ` +i2` p2+iQ`b `2im`M 2H2K2Mib rBi? K i+?BM; M K2bX h?Bb QMHv rQ`Fb B7 i?2 p2+iQ` Bb M K2/X

```
t l@+UkX-R9X-kjXj- 8XV9

O TTHv M K2b
M K2bUbV l@U] ]- ]#]- ]+]- ]/]V

O bm#b2i mbBM; M K2b
t(+U]/]- ]+]- ] ]V)
O= / +
O= 8X9 jXj kXR

O GBF2 BMi2;2` BM/B+2b- vQm + M `2T2 i BM/B+2b
t(+U] ]- ] ]- ] ]V)
O=
O= kXR kXR kXR

O * `27mH5 L K2b `2 Hr vb K i+?2/ 2t +iHv
t l@+U#+ 4R- /27 4kV
t(+U] ]- ]/]V)
O= lL = lL =
O= L L
```

9X GQ;B+ H p2+iQ`b b2H2+i 2H2K2Mib r?2`2 i?2 +Q``2bTQM/BM; HQ;B+ H p2+iQ` Bb hh_l1X

```
t l@+UkX-R9X-kjXj- 8XV9
t(+Uh_l1- h_l1- 6 Ga1- 6 Ga1V)
```

```
O= (R) kXR 9Xk
```

3XRXk *QM/BiBQM H am#b2iiBM;

GQ;B+ H bm#b2iiBM; Bb i?2 KQbi mb27mH ivT2 Q7 bm#b2iiBM;- #2
bm#b2i # b2/ QM BiBQM HQ;B+ X

h?2 UHQ;B+ HV +QKT `BbQM QT2` iQ`b FMQrM iQ _ `2,

Ç l 7Q` H2bb i? M
Ç =7Q` ;`2 i2` i? M
Ç l47Q` H2bb i? M Q` 2[m H iQ
Ç =47Q` ;`2 i2` i? M Q` 2[m H iQ
Ç 447Q` 2[m H iQ 2 +? Qi?2`
Ç 54MQi 2[m H iQ 2 +? Qi?2`

h?2 MB+2 i?BM; #Qmi _ Bb i? i vQm + M mb2 i?2b2 +QKT `BbQM Q
p2+iQ`bX 6Q` 2t KTH2,

```
t l@+UkXR9XkjXj- 8XV9
t =j
O= (R) 6 Ga1 h_l1 h_l1 h_l1
```

h?Bb +QKK M/ i2bib 7Q` 2p2`v 2H2K2Mi Q7 i?2 p2+iQ` B7 i?2 +QM
i?2 +QKT `BbQM QT2` iQ`X 6Q` 2 +? M/ Bi `2im`Mb HQ;B+ H p2+iQ`5

q2 + M MQr T bb i?Bb bi i2K2Mi #2ir22M i?2 b[m `2 #` +F2ib i? i 7
bm#b2i QMHv i?Qb2 Bi2Kb i? i K i+?

```
t(t =j)
O= (R) 9Xk jXj 8X9
```

uQm + M +QK#BM2 +QM/BiBQMH bm#b2iiBM; rBi? Q

```
t l@+UkXR9XkjXj- 8XV9

O +QK#BM; irQ +QM/BiBQM H bi i2K2Mib rBi?
t =j t l 8
O= (R) 6 Ga1 h_l1 h_l1 6 Ga1
t(t =j t l 8)
O= (R) 9Xk jXj

O +QK#BM; irQ +QM/BiBQM H bi i2K2Mib rBi? %
t l j %t =8
O= (R) h_l1 6 Ga1 6 Ga1 h_l1
t(t l j %t =8)
O= (R) kXR 8X9
```

```
O +QK#BMBM; +QM/BiBQM H bi i2K2Mib rBi? 5
5t = 8
O= (R) h_l1 h_l1 h_l1 6 Ga1
t(5t = 8)
O= (R) kXR 9Xk jXj
```

MQi?2` r v iQ ;2M2` i2 BKTHB+Bi +QM/BiBQMbWBiMQWi2i2@K2Mib Bb mbBM; i?2 iQ`- r?B+? i2bib r?2i?2` M Bi2K Bb BM b2i,

```
t I@+UkX+R9X+kjXj- 8XV9

O ;2M2` i2 BKTHB+Bi HQ;B+ H p2+iQ`b i?`Qm;? i?2 WBMW QT2` iQ`
t WBMW+UjXj- 9XVk
O= (R) 6 Ga1 h_l1 h_l1 6 Ga1
t(t WBMW+UjXj- 9XVk)
O= (R) 9Xk jXj
```

3XRXj *? HH2M;2

am#b2+Q mMi`vXp2+iQ` r iQ `2im`M 2p2`v p Hm2 1s*1Sh ó* M / ô M/ ó"` xBHô

```
+QmMi`vXp2+UQ`I@ MBbi- M] M /-]]aB2`` G2-QM2]MK`-F]]C T M]"` xBMV]

O .Q Bi mbBM; TQbBiBp2 BMi2;2`b

O .Q Bi mbBM; M2; iBp2 BMi2;2`b

O .Q Bi mbBM; HQ;B+ H p2+iQ`

O .Q Bi mbBM; +QM/BiBQM H bi i2K2Mi U M/ M BKTHB+Bi HQ;B+ H p2
```

3Xk am#b2iiBM; GBbib

am#b2iiBM; HBbib rQ`Fb BMi?2 b K2 r v b bm#b2iiBM; M iQKB+ p2+iQ`X > i?2`2öb QM2 BKTQ`i M(i r/BHr2`2Hr+2, `2im`M( M HBbiX /2b+`B#2/ #2HQr- H2i vQm TmHH Qmi i?2 +QKTQM2Mib Q7 i?2 HBbiX

G2iöb BHHmbi` i2 rBi? i?2 7QHHQrBM; HBbi

```
KvnHBbi I@UHBbi 4 Rj- # 4] bi`BM;]+ 4TB-/ 4 HBbiU@R- @8V
```

3XkXR q(Bi?

(2ti` +ib bm#@HBbi r?2`2 i?2 `2bmHi rBHH Hr vb #2 HBbiX GI
vQm + M bm#b2i rBi? HQ;B+ H- BMi2;2`- Q` +? ` +i2` p2+iQ`X

```
KvnHBbi(R,k)
O= 0
O= (R) R k j
O=
O= 0#
O= (R) ] bi`BM;]
bi`UKvnHBbi(R,k)V
O= GBbi Q7 k
O= 0 , BMi (R,j) R k j
O= 0 #, +?` ] bi`BM;]

KvnHBbi(9)
O= 0/
O= 0/((R))
O= (R) @R
O=
O= 0/((k))
O= (R) @8
bi`UKvnHBbi(9)V
O= GBbi Q7 R
O= 0 /,GBbi Q7 k
O= XX0 , MmK @R
O= XX0 , MmK @8

KvnHBbi(]b])
O= 0
O= (R) R k j
bi`UKvnHBbi(]b])V
O= GBbi Q7 R
O= 0 , BMi (R,j) R k j
```

3XkXk q(Bi?

((2ti` +ib bBM;H2 2H2K2Mib HBbiX AM Qi?2` rQ`/b- Bi `2KQp2b i?
?B2` `+?v M/ `2im`Mb r? i2p2` Q#D2+i Bb biQ`2/ BMbB/2X

```
KvnHBbi((R))
O= (R) R k j
bi`UKvnHBbi((R))V
O= BMi (R,j) R k j
```

```
O +QKT `2 iQ
KvnHBP)i(
O= 0
O= (R) R k j
bi`UKvnHBR)bM(
O= GBbi Q7 R
O= 0 , BMi (R,j) R k j
```

h?2 /BbiBM+iBQM M#2(rB2M`2 HHv BKTQ`i Mi 7Q(/HBbi #2+ mb2 /QrM BMiQ i?2 H`2im?BMb2 M2r- bK HH2` HBbiX

óA7 HBbi i` BM + ``vBM; Q(D8)+b-i??2QM#D2+i BM + ` 8d(9,e)Bb i` BM Q7 + `b 9@eXô

\

3XkXj rB0i?

0Bb b?Q`i? M/ 7Q` 2ti` +iBM;M K2/ 2H2K2Mib Q7 HBbiX Ai rQ`Fb bBKB ((2t+2Ti i? ivQm /QMöi M22/ iQ mb2 [mQi2bX

```
KvnHBPbi
O= (R) R k j

O b K2 b
KvnHBbi()
O= (R) R k j
```

h?20QT2` iQ` #2+QK2b 2bT2+B HHv ?2HT7mH r?2M TTHB2/iQ/ i 7` K2b-2 KQ`2 #2HQrX

3XkX9 *? HH2M;2

h F2 HQQF i i?2 HBM2 ` KQ/2H #2HQr,

```
KQ/ I@UKT;ri- / i 4 Ki+ `bV
bmKKU`KQ/V
O=
O= * HH,
O= HKU7Q`KmH 4 KT; ri- / i 4 Ki+ `bV
O=
O= _2bB/m Hb,
O= JBM RZ J2/B M jZ J t
O= @9X89j @kXje8 @yXRk8 RX9Ry eX3dj
O=
O= *Q277B+B2Mib,
```

```
O=        1biBK i2 ai/X 1``Q` i p Hm2 S`U=%i%V
O= UAMi2`+2TiV  jdXR X3d3  RNX3e  I k2@Re
O= ri         @8Xj99  yX88N   @NX8e  RXj2@Ry
O= @@@
O= aB;MB7X +Q/2b,  y ^   ^ yXyyR ^  ^ yXyR ^ ^ yXy8 ^X^ yXR ^
O=
O= _2bB/m H bi M/ `/ 2``Q`, jXy8 QM jy /2;`22b Q7 7`22/QK
O= JmHiBTH2 _@b[m `2/,  yXd8j-   /Dmbi2/ _@b[m `2/,  yXd98
O= 6@bi iBbiB+, NRX9 QM R  M/ jy .6-  T@p Hm2, RXkN2@Ry
```

1ti` +i i?2 _ b[m `2/ 7`QK i?2 KQ/2H bmKK `vX

3Xj am#b2iiBM; J i`B+2b

aBKBH ` iQ p2+iQ`b- vQm + M mb2)?Q b2H2+iQ`b iQ bm#b2i M2 i`BK mHiBTH2 2H2K2Mib 7`QK  K i`BtX "mi r?2`2 b p2+iQ`b ? p2 QM2 /BK2MbBQM irQ /BK2MbBQMbX q2 i?2`27Q`2 ? p2 iQ mb2 irQ bm#b2iiBM; p2+iQ iQ b2H2+i- MQi?2` 7Q` +QHmKMb b2T ` i2/ #v +QKK X

*?2+F Qmi i?2 7QHHQrBM; K i`Bt,

```
 I@K i`BUR,N- M`Qr j4V
+QHM UK 2Vb I@U] ]- ]"]- ]*]V

O=      " *
O= (R-) R 9 d
O= (k-) k 8 3
O= (j-) j e N
```

q2 + M bm#b2i i?Bb K i`Bt #v T bbBM; irQ bm#b2iiBM; p2+iQ`b, QM MQi?2` iQ b2H2+i +QHmKMb,

```
O b2H2+ib i?2 p Hm2  i i?2 7B`bi `Qr  M/ b2+QM/ +QHmKM
 (R- k)
O= "
O= 9

O b2H2+ib 7B`bi `Qr-  M/ i?2 7B`bi  M/ i?B`/ +QHmKMb
 (R- @)k
O=    *
O= R d

O b2H2+ib 7B`bi irQ `Qrb-  M/ i?2 7B`bi  M/ i?B`/ +QHmKMb
 (+UR- kV- +UR- jV)
O=      *
```

```
O= (R-) R d
O= (k-) k 3
```

"H MF bm#b2iiBM; Bb HbQ mb27mH #2+ mb2 Bi H2ib vQm F22T HH `Qrb Q

```
(+UR- kV- )O b2H2+ib 7B`bi irQ `Qrb M/ HH +QHmKMb
O= " *
O= (R-) R 9 d
O= (k-) k 8 3
```

3X9 am#b2iiBM; . i 7` K2b

. i 7`QK/ i 7` K2b + M #2 //`2bb2/ HBF2 K i`B+2b- mbBM; irQ p2+iQ`b b
` i2/ #v +QKK X

```
O .27BMBiBQM Q7 p2+iQ`b
M K2 I@ +]J2`+m`v]- ]o2Mmb]- ] `i?]- ]J `b]- ]CmTBi2`]- ]a im`M]- ]l` Mmb]- ]L2TimM2]V
ivT2 I@ +]h2``2bi`B H TH M2i]- ]h2``2bi`B H TH M2i]- ]h2``2bi`B H TH M2i]- ]h2``2bi`B H TH
/B K2i2` I@ +UyXj3- kyXN9R- yX8jk- RRXkyN- NX99N- 9Xyyd- jX33jV
`BM;b I@ +U6 Ga1- 6 Ga1- 6 Ga1- 6 Ga1- h_l1- h_l1- h_l1- h_l1V

TH M2ib /l@ X7` K2UM K2- ivT2- /B K2i2`- `BM;bV
TH M2ib
O= M K2 ivT2 /B K2i2` `BM;b
O= R J2`+m`v h2``2bi`B H TH M2i yXj3k 6 Ga1
O= k o2Mmb h2``2bi`B H TH M2i yXN9N 6 Ga1
O= j 1 `i? h2``2bi`B H TH M2i RXyyy 6 Ga1
O= 9 J `b h2``2bi`B H TH M2i yX8jk 6 Ga1
O= 8 CmTBi2` : b ;B Mi RRXkyN h_l1
O= e a im`M : b ;B Mi NX99N h_l1
O= d l` Mmb : b ;B Mi 9Xyyd h_l1
O= 3 L2TimM2 : b ;B Mi jX33j h_l1
```

G2iöb i`v bQK2 bm#b2iiBM; MQrX

```
O S`BMi Qmi /B K2i2` Q7 J2`+m`v U`Qr R- +QHmKM jV
TH M2ib(R- j)
O= (R) yXj3k

O S`BMi Qmi / i 7Q` J `b U2MiB`2 7Qm`i? `QrV
TH M2ib(9- )
O= M K2 ivT2 /B K2i2` `BM;b
O= 9 J `b h2``2bi`B H TH M2i yX8jk 6 Ga1

O S`BMi 7B`bi irQ `Qrb Q7 i?2 7B`bi irQ +QHmKMb
```

```
TH M2Rb(Rk, Rk)
O= M K2 ivT2
O= R J2`+m`v h2``2bi`B H TH M2i
O= k o2Mmb h2``2bi`B H TH M2i
```

3X9XR am#b2iiBM; L0K2b M/

AMbi2 / Q7 mbBM; MmK2`B+b iQ b2H2+i 2H2K2Mib Q7 / i 7` K2- i?2 p `B #H2 M K2b iQ b2H2+i +QHmKMb Q7 / i 7` K2X

amTTQb2 vQm r Mi iQ b2H2+i i?2 }`bi i?`22 2H2K2Mib Q7 i?2 ivT2 r v iQ /Q i?Bb Bb

```
TH M2Rbj( k)
O= (R) ]h2``2bi`B H TH M2i] ]h2``2bi`B H TH M2i] ]h2``2bi`B H
```

TQbbB#H2 /Bb /p Mi ;2 Q7 i?Bb TT`Q +? Bb i? i vQm ? p2 iQ FMQ i?2 +QHmKM MmK#2` Q7 ivT2- r?B+? ;2ib ? `/ B7 vQm ? p2 HQi Q7 Q7i2M 2 bB2` iQ Dmbi K F2 mb2 Q7 i?2 p `B #H2 M K2,

```
TH M2Rbj( ]ivT2])
O= (R) ]h2``2bi`B H TH M2i] ]h2``2bi`B H TH M2i] ]h2``2bi`B H
```

uQm rBHH Q7i2M r Mi iQ b2H2+i M 2MiB`2 +QHmKM- M K2Hv QM2 b / i 7` K2X A7 vQm r Mi iQ b2H2+i HH 2H2K2Mib Q7 i?2 p `B #H2 2t KTH2- #Qi? Q7 i?2b2 rBHH /Q i?2 i`B+F,

```
TH M2ib)(-
O= (R) yXj3k yXN9N RXyyy yX8jk RRXkyN NX99N 9Xyyd jX3
TH M2ib([B K2i2`])
O= (R) yXj3k yXN9N RXyyy yX8jk RRXkyN NX99N 9Xyyd jX3
```

>Qr2p2`- i?2`2 Bb b?Q`i@+miX A7 vQm` +QHmKM b ? p2 M K2b- v bB;M,

```
TH M2ibO/B K2i2`
O= (R) yXj3k yXN9N RXyyy yX8jk RRXkyN NX99N 9Xyyd jX3
```

_2K2K#2` i? i / i b2ib `2 `2 HHv HBbib Q7 p2+iQ`b UQM2 p2+iQ` T2 bHBbi0Mbk2H2+iMi?2H2K2Mi 7`QK/i?2HBbi b2H2+iMi?2 +QHmKM Up2+iQ`V 7`QK i?2 / i 7` K2X

3X9Xk *QM/BiBQM H am#b2iiBM;

q? i B7 r2 r Mi iQ bm#b2i i?2 / i b2i # b2/ QM bQK2 +QM/BiBQM\ G r Mi iQ 2ti` +i HH i?2 TH M2ib rBi? /B K2i2` ;`2 i2` i? M j\ q2 +Qm

i?2 / i b2i M/ `2+Q`/ HH i?2 Q#b2`p iBQMb i? i }i i? i /2b+`BTiBQM- #mi i
i2/BQmb M/ 2``Q` T`QM2X

h?2`2öb #2ii2` r v5 q2 + M +QK#BM2 irQ TQr2`7mH bm#b2iiBM; iQQHb,
QT2` iQ` M/ +QM/BiBQM H bm#b2iiBM;X

6B`bi- r2 2ti` +i i?2 K2i2`+QHmKMX

```
/B K2i2`b T@ M/B K2i2`
```

h?2M- r2 }M/ i?2 2H2K2Mib i? i `2 ;`2 i2` i? M jX

```
/B K2i2`b j
O= (R) 6 Ga1 6 Ga1 6 Ga1 6 Ga1 h_l1 h_l1 h_l1 h_l1
```

Aiöb #QQH2 M p2+iQ`5 q2 + M (M-Q`) mQb2 ti?B+b BHH bBT/2 Mi2b
rBi?B K2i2` X j

h?BMF, `2 r2 bm#b2iiBM;b `Qr Q` +QHmKMb ?2`2\

```
TH M2ib(/B K2i2`b j- )
O= M K2 ivT2 /B K2i2` `BM;b
O= 8 CmTBi2` : b ;B Mi RRXkR h_l1
O= e a im`M : b ;B Mi NX98 h_l1
O= d l` Mmb : b ;B Mi 9XyR h_l1
O= 3 L2TimM2 : b ;B Mi jX33 h_l1

O b K2 b
O TH M2ib(TH M2ib0/B K2i2` = j- )
```

"2+ mb2 Bi HHQrb vQm iQ 2 bBHv +QK#BM2 +QM/BiBQMb 7`QK KmHiBTH2 +
bm#b2iiBM; Bb T`Q# #Hv i?2 KQbi +QKKQMHv mb2/ i2+?MB[m2 7Q` 2ti` +
Qmi Q7 / i 7` K2X

3X9Xj GBbi@GBF2 M/ J i`Bt@GBF2 am#b2iiBM;

. i 7` K2b TQbb2bb i?2 +? ` +i2`BbiB+b Q7 #Qi? HBbib M/ K i`B+2b, B7 v
rBi? bBM;H2 p2+iQ`- i?2v #2? p2 HBF2 HBbib- M/ `2im`M QMHv i?2 +QHm

```
/7 I@ i X7` K2(t 4 9,e- v 4j,R- x 4H2ii2`b(j)V

O GBF2 HBbi,
/7(+U]t]- ]x]V)
O= t x
O= R 9
O= k 8 #
O= j e +

O GBF2 K i`Bt
```

```
/7(-+U[t]- ]x]V)
O=   t x
O= R 9
O= k 8 #
O= j e +
```

"mi i?2`2öb M BKTQ`i Mi /Bz2`2M+2 r?2M vQm b2H2+i bBM;H2 +
bm#b2iiBM; bBKTHB}2b #v /27 mHi- HBbi bm#b2iiBM; /Q2b MQiX

```
/7 l@/ i X7` K!2 4 9,e- v 4j,R- x 4H2ii2`b([)V

O HBF2   HBbi
/7([t])
O=   t
O= R 9
O= k 8
O= j e
+H bb U/7([t])V
O= (R) ]/ i X7` K2]

O HBF2   K i`Bt
/7(-]t])
O= (R) 9 8 e
+H bb U/7(-]t])V
O= (R) ]BMi2;2`]
```

3X9X9 *? HH2M;2b

*? HH2M;2 RX

6Bt 2 +? Q7 i?2 7QHHQrBM; +QKKQM / i 7` K2 bm#b2iiBM; 2``Q`b

```
O +?2+F Qmi r? i r2^`2 /2 HBM; rBi?
Ki+ `b

O 7Bt
Ki+ `b(Ki+0+`bH 4- )
Ki+ `b(0R 9- )
Ki+ `b(Ki+0+`bH 4 8)
Ki+ `b(Ki+0+`bH 4 4 9 %e- )
```

*? HH2M;2 kX

q?v /Q2Ki+ `b(R,ky) 2`m`M M 2``Q`\ >Qr /Q2b Bi /Bz2` 7`QK i?2 bBKBH ` Ki+ `b(R,ky-)

3X8 am#@ bbB;MK2Mi

3X8XR " bB+b Q7 am#@ bbB;MK2Mi

HH bm#b2iiBM; QT2` iQ`b + M #2 +QK#BM2/ rBi? bbB;MK2Mi iQ KQ/B7v p Hm2b Q7 i?2 BMTmi p2+iQ`X

```
t I@R,8
t(+UR- kV) I@k,j
t
O= (R) k j j 9 8
```

h?Bb Bb 2bT2+B HHv mb27mH r?2M +QM/BiBQM HHv KQ/B7vBM; p2+iQ`b H2iöb b v r2 r Mi2/ iQ `2TH +2 HH p Hm2b H2bb i? M j rBi? L X

```
t I@R,8
t(t I j) I@L
t
O= (R) L L j 9 8
```

h?Bb HbQ rQ`Fb QM / i 7` K2bX G2iöb b v r2 r Mi2/ iQ KQ/B7v Qm` / i 7` K2X

```
O .27BMBiBQM Q7 p2+iQ`b
M K2 I@+ (J2`+m`v- o2Mmb- 1 `i?- J `b- CmTBi2`- a im`M- l` Mmb- L2TimM2)
ivT2 I@+ (h2``2bi`B H TH M2i- h2``2bi`B H TH M2i- h2``2bi`B H TH M2i- h2``2bi`B H TH
/B K2i2` I@+ (yXj3k- yXN9N- R- yX8jk- RRXkyN- NX99N- jXyyd- jX33j)
`BM;b I@+ (6 Ga1- 6 Ga1- 6 Ga1- 6 Ga1- h_l1- h_l1- h_l1- h_l1)

TH M2ib I@/ i X7` K2(M K2- ivT2- /B K2i2`- `BM;b- bi`BM;bb6 +iQ`b 4 6V
TH M2ib
O= M K2 ivT2 /B K2i2` `BM;b
O= R J2`+m`v h2``2bi`B H TH M2i yXj3k 6 Ga1
O= k o2Mmb h2``2bi`B H TH M2i yXN9N 6 Ga1
O= j 1 `i? h2``2bi`B H TH M2i RXyyy 6 Ga1
O= 9 J `b h2``2bi`B H TH M2i yX8jk 6 Ga1
O= 8 CmTBi2` : b ;B Mi RRXkyN h_l1
O= e a im`M : b ;B Mi NX99N h_l1
O= d l` Mmb : b ;B Mi 9Xyyd h_l1
O= 3 L2TimM2 : b ;B Mi jX33j h_l1
```

G2iöb b v r2 r Mi iQ `2TH +2 i?2 i2`K óh2``2bi`B H TH M2iô rBi? óh
M22/ iQ bm#bi2TQ` i?Qb2 2H2K2Mib,

```
TH M2ivT2 4 4]h2``2bi`B H TH M2i]
O= (R) h_l1 h_l1 h_l1 h_l1 6 Ga1 6 Ga1 6 Ga1 6 Ga1
```

LQr r2 + M `2@ bbB;M i?2 p Hm2b Q7

```
TH M2ivT2(TH M2ivT2 4 4]h2``2bi`B H TH M2i]
O= (R) ]h2``2bi`B H TH M2i] ]h2``2bi`B H TH M2i] ]h2``2bi`B H T
O= (9) ]h2``2bi`B H TH M2i]
TH M2ivT2(TH M2ivT2 4 4]h2``2bi`B H TH M2i] (@]hS]
TH M2ib
O= M K2 ivT2 /B K2i2` `BM;b
O= R J2`+m`v hS yXj3k 6 Ga1
O= k o2Mmb hS yXN9N 6 Ga1
O= j 1 `i? hS RXyyy 6 Ga1
O= 9 J `b hS yX8jk 6 Ga1
O= 8 CmTBi2` : b ;B Mi RRXkyN h_l1
O= e a im`M : b ;B Mi NX99N h_l1
O= d l` Mmb : b ;B Mi 9Xyyd h_l1
O= 3 L2TimM2 : b ;B Mi jX33j h_l1
```

3X8Xk _2+v+HBM;

q?2M TTHvBM; M QT2` iBQM iQ irQ p2+iQ`b i? i `2[mB`2b i?2K iQ
H2M;i?- _ miQK iB+ HHv `2+v+H2b- Q` `2T2 ib- i?2 b?Q`i2` QM2
2MQm;? iQ K i+? i?2 HQM;2` QM2X

```
/7 l@ i X7` Ul2 4 9, d- v 4 H2ii2`b(R,9)V

O ` `2+v+H2b p Hm2b
/7 0t l@+UR- kV
/7
O= t v
O= R R
O= k k #
O= j R +
O= 9 k /

O bQK2iBK2b i?Bb Bb ?2HT7mH B7 vQm r Mi iQ `2TH +2 M 2MiB
/7 0t l@/7 0t Yj
/7
O= t v
O= R 9
O= k 8 #
```

```
O= j 9 +
O= 9 8 /
```

3X8Xj TTHB+ iBQMb

h?2 # bB+ T`BM+BTH2b /2b+`B#2/ #Qp2 ;Bp2 `Bb2 iQ rB/2 p `B2iv Q7 mb2
iBQMbX aQK2 Q7 i?2 KQbi BKTQ`i Mi TTHB+ iBQMb `2 /2b+`B#2/ #2HQ
Q7 i?2b2 # bB+ i2+?MB[m2b `2 r` TT2/ mT BMiQ KQ`2 +QM+Bb2 7mM+iB
bm#b2iU2V`;2UTV Hv`,, `` MV2-#Vmi Bi Bb mb2 7mH iQ mM/2`bi M/ ?Qr i?2v
`2 BKTH2K2Mi2/ rBi?# bB+ bm#b2iiBM;X h?Bb rBHH HHQr vQm iQ / Ti iQ
m iBQMb i? i `2 MQi /2 Hi rBi? #v 2tBbiBM; 7mM+iBQMbX

P`/2`BM; *QHmKMb

*QMbB/2` r2 ? p2 i?Bb / i 7` K2,

```
/7 I@ i X7` K U2
  *QmMi`v +U4 A` [ - ]*?BM - ]J2tB+Q - ]_mbbB - ]lMBi2/ EBM;/QK] V-
  _2;BQM +U4 ]JB//H2 1 bi - ]]bB - ]]LQ`i? K2`B+ - ]1]bi2`M 1m`QT2 - ]q2bi2`M 1m`QT2] V-
  G M;m ;2 +U4 ` #B+ - ]]J M/ `BM - ]aT MBb? - ]_mbbB M - ]1M;HBb?] V
V
/7
O=        *QmMi`v      _2;BQM  G M;m ;2
O= R        A` [    JB//H2 1 bi     ` #B+
O= k       *?BM          bB  J M/ `BM
O= j       J2tB+Q  LQ`i?  K2`B+   aT MBb?
O= 9      _mbbB  1 bi2`M 1m`QT2  _mbbB M
O= 8 lMBi2/ EBM;/QK q2bi2`M 1m`QT2  1M;HBb?
```

q? i B7 r2 r Mi2/ iQ `2Q`/2` i?2 +QHmKMb }bi?q2 + M /Q _2;BQM- i?2
bQ mbBM; bm#b2iiBM; rBi? i?2 M K2b UQ` MmK#2`V Q7 i?2 +QHmKMb,

```
/7 I@ i X7` K U2
  *QmMi`v +U4 A` [ - ]*?BM - ]J2tB+Q - ]_mbbB - ]lMBi2/ EBM;/QK] V-
  _2;BQM +U4 ]JB//H2 1 bi - ]]bB - ]]LQ`i? K2`B+ - ]1]bi2`M 1m`QT2 - ]q2bi2`M 1m`QT2] V-
  G M;m ;2 +U4 ` #B+ - ]]J M/ `BM - ]aT MBb? - ]_mbbB M - ]1M;HBb?] V
V

O `2Q`/2` +QHmKMb mbBM; M K2b
M K2bU/7V
O= (R) ]*QmMi`v] ]_2;BQM] ]G M;m ;2]
/7R I@ /7( - +U]_2;BQM] - ]*QmMi`v] - ]G M;m ;2] V)
/7R
O=         _2;BQM    *QmMi`v G M;m ;2
```

```
O= R    JB//H2 1 bi    A` [    ` #B+
O= k         bB         *?BM  J M/ `BM
O= j  LQ`i?  K2`B+   J2tB+Q  aT MBb?
O= 9 1 bi2`M 1m`QT2_mbbB   _mbbB M
O= 8 q2bi2`M 1m`QT2 lMBi2/ EBM;/QK  1M;HBb?

O `2Q`/2` +QHmKMb mbBM; BM/B+2b
M KUb U/7V
O= (R) ]*QmMi`v]  ]_2;BQM]   ]G M;m ;2]
/7R I@7(-+Uk-R-jV)
/7R
O=        _2;BQM    *QmMi`v G M;m ;2
O= R    JB//H2 1 bi    A` [    ` #B+
O= k         bB         *?BM  J M/ `BM
O= j  LQ`i?  K2`B+   J2tB+Q  aT MBb?
O= 9 1 bi2`M 1m`QT2_mbbB   _mbbB M
O= 8 q2bi2`M 1m`QT2 lMBi2/ EBM;/QK  1M;HBb?
```

PM2 ?2HTmH 7mM+iBQM Q`/2`UV BM S M/ b +QMbi`m+ib  MX Aii F2b  p2+iQ` b BMTmi M/ `2im`Mb M BMi2;2` p2+iQ` /2b+`B#BM; ?Qr i?2 bm#b2ii2/ p2+iQ` b?QmH/ #

```
t I@+U]#]- ]+]- ] ]V
Q`/2`UtV
O= (R) j R k
t(Q`/2`UtV)
O= (R) ] ] ]#] ]+]
```

EMQrBM; i?Bb- r2 +M mb2 Q`/2` Qm` +QHmKMb #v  HT? #2iB+ H Q`

_2KQpBM; UQ` F22TBM;V +QHmKMb 7`QK / i 7` K2bX

h?2`2  `2 irQ r vb iQ `2KQp2 +QHmKMb 7`QK / i 7` K2X uQm + M b2 +QHmKMb LlGG:

```
/7 I@ / i X7` K2U
  *QmMi`v+U]A` []- ]*?BM]- ]J2tB+Q]- ]_mbbB ]- ]lMBi2/ EBM;/QK]V-
  _2;BQM+U]JB//H2 1 bi]- ]bB ]- ]LQ`i?  K2`B+ ]- ]1 bi2`M 1m`QT2]- ]q2bi2`M 1m`QT2]V-
  G M;m ;2+U] ` #B+ ]- ]J M/ `BM]- ]aT MBb?]- ]_mbbB M]- ]1M;HBb?]V
V

/70G M;m ;2 LlI@ G
```

P` vQm + M bm#b2i iQ `2im`M QMHv i?2 +QHmKMb vQm r Mi,

```
/7 I@ / i X7` K2U
  *QmMi`v+U]A` []- ]*?BM]- ]J2tB+Q]- ]_mbbB ]- ]lMBi2/ EBM;/QK]V-
  _2;BQM+U]JB//H2 1 bi]- ]bB ]- ]LQ`i?  K2`B+ ]- ]1 bi2`M 1m`QT2]- ]q2bi2`M 1m`QT2]V-
```

```
  G M;m ;2+U4]` #B+]]J M/ `B-M]aT MB-b?]_mbbB-[M1]M;HBVb?]
V

/7R l@7(-+U]*QmMi-`v]_2;BQVM]
/7R
O=        *QmMi`v      _2;BQM
O= R         A` [     JB//H2 1 bi
O= k        *?BM           bB
O= j       J2tB+Q  LQ`i?  K2`B+
O= 9       _mbbB  1 bi2`M 1m`QT2
O= 8 lMBi2/ EBM;/QK q2bi2`M 1m`QT2

O mbBM; M2; iBp2 BMi2;2`b
/7k l@7(-@)j
/7k
O=        *QmMi`v      _2;BQM
O= R         A` [     JB//H2 1 bi
O= k        *?BM           bB
O= j       J2tB+Q  LQ`i?  K2`B+
O= 9       _mbbB  1 bi2`M 1m`QT2
O= 8 lMBi2/ EBM;/QK q2bi2`M 1m`QT2
```

*? Ti2` N

qQ`FBM; rBi? . i

h?Bb mMBi rBHH +Qp2` i?2 # bB+b Q7 rQ`FBM; rBi? / i -BM+Hm/BM; T`QD2
/ i i2`Kb M/ +QM+2Tib- BKTQ`iBM; / i - M/ 2tTHQ`BM; / i X

NXR S`QD2+i qQ`F~Qr

PM2 / v

Ç vQm rBHH M22/ iQ [mBi _- ;Q /Q bQK2i?BM; 2Hb2- M/ `2im`M iQ vQm`
i?2 M2ti / vX
Ç vQm rBHH #2 rQ`FBM; QM KmHiBTH2 T`QD2+ib bBKmHi M2QmbHv- M/
iQ F22T i?2K b2T ` i2X
Ç vQm rBHH M22/ iQ #`BM; / i 7`QK i?2 QmibB/2 rQ`H/ BMiQ _- M/ b2M/
K2`B+ H `2bmHib M/ };m`2b 7`QK _ # +F QmiBMiQ i?2 rQ`H/X

h?Bb mMBi rBHH i2 +? vQm ?Qr iQ b2i mT vQm` rQ`F~Qr iQ K F2 i?2 #2bi mb

NXRXR aiQ`2 M Hvb2b BM a+`BTib- LQi qQ`FbT +2bX

_ aim/BQ miQK iB+ HHv T`2b2`p2b vQm` rQ`FbT +2 U2MpB`QMK2Mi M/ +
?BbiQ`vV r?2M vQm [mBi _- M/ `2@HQ /b Bi i?2 M2ti b2bbBQMX A `2+QK
im`M i?Bb #2? pBQ` QzX

h?Bb rBHH + mb2 vQm bQK2 b?Q`i@i2`K T BM- #2+ mb2 MQr r?2M
/BQ- Bi rBHH MQi `2K2K#2` i?2 `2bmHib Q7 i?2 +Q/2 i? ivQm ` M H
b?Q`i@i2`K T BM rBHH b p2 vQm HQM;@i2`K ;QMv- #2+ mb2 Bi 7Q
HH BKTQ`i Mi BMi2` +iBQMb BM vQm` b+`BTibX

NXRXk qQ`FBM; .B`2+iQ`B2b M/ S i?b

GBF2 K Mv T`Q;` KKBM; H M;m ;2b- _ ? b T`Q`2F`BM;H MQiBQM Q7
/B`2+iQ`v X h?Bb Bb r?2`2 _ HQQFb 7Q` }H2b i? ivQm bF Bi iQ HQ /-
rBHH Tmi Mv }H2b i? ivQm bF Bi iQ b p2X

_aim/BQ b?Qrb vQm` +m``2Mi rQ`FBM; /B`2+iQ`v i i?2 iQT Q7 i?2
+ MT`BMi i?Bb Qmi BM _ +;2ir/UV `mMMBM;

```
;2ir/UV
O= (R) ]fIb2`bf`i2`K Mf.2bFiQTf+Qm`b2]
```

A /Q MQi `2+QKK2M/ Bi- #mi vQm + M b2i i?2 rQ`FBM; /B`2+iQ`v 7`

```
b2ir/U]fT i?fiQfKvf*QQHS V QD2+i]
```

h?2 +QKK M/ #Qp2 T`Bmi?bQvQm` rQ`FBM; /B`2+iQ`vX h?BMF Q7
T i? b M //`2bbX S i?b `2 BM+`2/B#Hv BKTQ`i Mi BM T`Q;` KK
#2 HBiiH2 i`B+FvX G2iöb ;Q BMiQ #Bi KQ`2 /2i BHX

#bQHmi2 S i?b

#bQHmi2 T i?b `2 T i?b i? i TQBMi iQ i?2 b K2 TH +2 `2; `/H2bb Q7 vQm
`2Mi rQ`FBM; /B`2+iQ`vX h?2v Hr vb TQBMi iQ i?2 /B`2+iQ`v i?2v `2 BM i?2 H/b
2p2`vi?BM; 2Hb2 QM vQm` +QKTmi2`X

Ç AM qBM/Qrb- #bQHmi2 T i?b bi `i rBi? *,V `Q` 2i+X U2X;X
+FbH b?2b `$bb2Xp;2` MXK2

Ç AM J +fGBMmt i?2v bi `i rBi? fX i?2 H2 /BM; bH b? BM
fmb2`bf`i2XK M

AMbB/2 i?2 `QQi /B`2+iQ`v `2 b2p2` H Qi?2` /B`2+iQ`B2b- r?B+? r2 + H
`2+iQ`B2b FMQrM i? i i?2 /B`2+iQ`v `BM i?2 /B`2+iQ`v KB2
#2+ mb2 QM2 i?2 }`bi T `i Q7 Bib M K2X aBKBH`Hv- r2 FMQr i? i
biQ`2/ BMbB/2 i?2 `QQi /B`2+iQ`v BM K2 #X;BMb rBi?

LQiB+2 i? i i?2`2 `2 irQ K2 MBM;b 7Q` i?2 M Bi
TT2 `b i i?2 7`QMi Q7 }H2 Q` /B`2+iQ`v M K2- = Bi `272`b iQ i?2 `QQi
/B`2+iQ`vX q?2M Bi TT2 `b- = Bi`b Dmbi b2T ` iQ`X

J +fGBMmt pbX qBM/Qrb

h?2`2 `2 irQ # bB+ bivH2b Q7 T i?b, J +fGBMmt M/ qBM/QrbX h?2 K B
/Bz2`2M+2 Bb ?Qr vQm b2T ` i2 i?2 +QKTQM2Mib Q7 i?2 T i?X J + M/ G
mb2 bH b?2b- r?BH2 qBM/Qrb mb2b # +FbH b?2b U2X;X
THQib$/B KQMMX T/7

_ + M rQ`F rBi? 2Bi?2` ivT2- MQ K ii2` r? i TH i7Q`K vQmö`2 +m``2MiHv m
IM7Q`imM i2Hv- # +FbH b?2b K2 M bQK2i?BM; bT2+B H iQ _- M/ iQ ;2i
+FbH b? BM i?2 T i?- vQm M22/ iQ ivT2 irQ # +FbH b?2b5 h? i K F2b HB
i` iBM;- bQ A `2+QKK2M/ Hr vb mbBM; i?2 GBMmtfJ + bivH2 rBi? 7Q`r `/ bH

>QK2 .B`2+iQ`v

aQK2iBK2b vQmöHH b22 +? ` +i2` BM T i?X

Ç AM J +fGBMmt- Bi? 2+QMp2MB2Mi b?Q`i+mi 7Q` vQm` ?QK2 /B`2+iQ`v
UfmbIb2`bf`i2VKX M

Ç qBM/Qrb /Q2bMöi `2 HHv ? p2 i?2 MQiBQM Q7 ?QK2 /B`2+iQ`v-
Bi mbm HHv TQBMib iQ vQm` /Q+mK2Mib /B`2+iQ`v U
a2iiBM;b$`i2VK M

#bQHmi2 pbX _2H iBp2 S i?b

uQm b?QmH/ i`v MQi iQ mb2 #bQHmi2 T i?b BM vQm` b+`BTib- #2+ mb2 i?
b? `BM;, MQ QM2 2Hb2 rBHH ? p2 2t +iHv i?2 b K2 /B`2+iQ`v +QM};m` iB

MQi?2` r v iQ /B`2+i _ iQ bQK2i?BM; 2Hb2 iQ bQK2i?BM; `2H iBp2 X Bp2 Bi

_2H iBp2 T i?b TQBMi iQ bQK2i?BM; `2H iBp2 iQ r?2`2 vQm `2- ` i?2 `QQi Q7 i?2 }H2 bvbi2KX 6Q` 2t KTH2- B7 vQm` +m``2Mi rQ`F f?QK2f`i2`K Mf2 M i?2 `2H iBp2 T i? Bb +QMi Bp2 +ib iQ i?2 7mHH #bQHmi2 T i?; f?QK2f`i2`K Mf/ i fmM X+bp

NXRXj _ S`QD2+ib

b #2;BMMBM; _ mb2`- Biöb PE iQ H2i vQm` ?QK2 /B`2+iQ`v- /Q+m Q` Mv Qi?2` r2B`/ /B`2+iQ`v QM vQm` +QKTmi2` #2 _öb rQ`FBM; /

"mi 7`QK i?Bb TQBMi 7Q`r `/- vQm b?QmH/ #2 Q`; MBxBM; vQm` T + i2/ bm#/B`2+iQ`B2b- +QMi BMBM; HH i?2 }H2b bbQ+B i2/ rBi? / i - _ b+`BTib- `2bmHib- };m`2bX

h?Bb Bb bm+? +QKKQM T` +iB+2 i? i _aim/BQ ? b #mBHi@BM bm T`QD2+ibX

G2iöb K F2 T`QD2+i iQ 6Q HQr 2H ` = L2r S`QD2+i

h?BMF + `27mHHv #Qmi r?B+? bm#/B`2+iQ`v vQm Tmi i?2 T`QD2- biQ`2 Bi bQK2r?2`2 b2MbB#H2- Bi rBHH #2 ? `/ iQ }M/ BM i?2 7mim

PM+2 i?Bb T`Q+2bb Bb +QKTH2i2- vQmöHH ;2i M2r _aim/BQ T`QD ó?QK2ô /B`2+iQ`v Q7 vQm` T`QD2+i Bb i?2 +m``2Mi rQ`FBM; /B`2+

```
;2ir/UV
O= (R) ]fIb2`bf`i2`K Mf.2bFiQTf+Qm`b2]
```

LQr r?2M2p2` vQm `272` iQ }H2 rBi? `2H iBp2 T i?- Bi rBHH HQQ

:Q ?2 / M/ +`2 i2 M2r _ b+`BTi M/ b p2 Bi BMbB/2 i?2 T`QD2+i 7

ZmBi _aim/BQX AMbT2+i i?2 7QH/2` bbQ+B i2/ rBi? vQm` T`QD2+i MQ
X_T`QD }H2X .Qm#H2@+HB+F i? i }H2 iQ `2@QT2M i?2 T`QD2+iX LQiB+2
iQ r?2`2 vQm H27i Qz, Biöb i?2 b K2 rQ`FBM; /B`2+iQ`v M/ +QKK M/ ?BI
M/ HH i?2 }H2b vQm r2`2 rQ`FBM; QM `2 biBHH QT2MX "2+ mb2 vQm
Kv BMbi`m+iBQMb #Qp2- vQm rBHH- ?Qr2p2`- ? p2 +QKTH2i2Hv 7`2b? 2
;m ` Mi22BM; i? i vQmö`2 bi `iBM; rBi? +H2 M bH i2X

NXRX9 6BH2 P`; MBx iBQM

uQm b?QmH/ #2 b pBM; HH vQm` }H2b bbQ+B i2/ rBi? vQm` T`QD2+i BM Q
>2`2öb # bB+ Q`; MBx iBQM bi`m+im`2 i? i A `2+QKK2M/,

```
K bi2`bni?2bBb,
  K bi2`bni?2bBbX_T`QD
  yRn*H2 MX_
  yknJQ/2HX_
  yjnoBbm HBx iBQMbX_
  . i f
    ` rf
      mM@` rX+bp
      rQ`H/# MF@` rX+bp
    +H2 M2/f
      +QmMi`v@v2 `X+bp
  _2bmHib,
    `2;`2bbBQMb
      ?RXiti
      ?kXiti
    7B;m`2b
      #Bp `B i2XT/7
      # `nTHQiXT/7
```

>2`2 `2 bQK2 BKTQ`i Mi iBTb,

- `2 /` r/ i 7`QK i?2 bm#/B`2+iQ`vX .QMöi 2p2` +? M;2 Q` Qp2`r`Bi2 i?2 ` r/ i 5
- 2tTQ`i +H2 M2/ M/ Hi2`2/ / i BMiQ b2T ` i2 /B`2+iQ`vX
- r`Bi2 b2T ` i2 b+`BTib 7Q` 2 +? bi ;2 BM i?2 `2b2 `+? TBT2HBM2X E22
  b?Q`i M/ 7Q+mb2/ QM QM2 K BM Tm`TQb2X A7 b+`BTi ;2ib iQQ HQ
  KB;?i #2 bB;M vQm M22/ iQ bTHBi Bi mTX
- r`Bi2 b+`BTib i? i `2T`Q/m+2 vQm` `2bmHib M/ };m`2b- M/ r`Bi2 i?2K
  i?2_2bmHib #/B`2+iQ`vX

+FMQrH2/;2K2Mib

h?Bb T;2 Bb BMT `i /2`Bp2/ 7`QK i?2 7QHHQrBM; bQm`+2b,

RX_ 7Q` .i a+B2M+2Mb2/ *m2M/2`p2 *QKKQMb ii`B#miBQM@ H2M+2M/2`p2
LQM*QKK2`+B H@LQ.2`Bpb jXy

JQ`2 _2bQm`+2b

Ç :2MixFQr- J ii?2r M/ C2bb2 JX a? TB`QX kyR9X *Q/2 M/ .i 7
aQ+B H a+B2M+2b, S` +iBiBQM2`öb :mB/2X

NXk AMi`Q/m+iBQM iQ . i

h?2 mT+QKBM; r22Fb rBHH #2 7Q+mb2/ QM mbBM; _ 7Q` / i +H2 M
G2iöb }`bi ;2i QM i?2 b K2 T ;2 rBi? bQK2 i2`Kb,

Ç p `B #H2 [m MiBiv- [m HBiv- Q` T`QT2`iv i? i vQm + M K2 bm

Ç MQ#b2`p iBQM Mb2i Q7 K2 bm`2K2Mib 7Q` i?2 b K2 mMBiX M Q
iBQM rBHH +QMi BM b2p2` H p Hm2b- 2 +? bbQ+B i2/ rBi? /
AöHH bQK2iBK2b `272` iQ M Q#b2`p iBQM b / i TQBMiX

Ç p Hm2b i?2 bi i2 Q7 p `B #H2 7Q` T `iB+mH ` Q#b2`p iBQM

Ç h #mH ` / i Bb b2i Q7 p Hm2b- 2 +? bbQ+B i2/ rBi? p `B #H2 M
b2`p iBQMX h #mH ` / i ? b `Qrb UQ#b2`p iBQMbV M/ +QHmK
HbQ + HH2/ iB/v / i Q` T`2 /b?22ib

NXkXR q?2`2öb Kv / i \

hQ bi `i- vQm }`bi M22/ iQ FMQr r?2`2 vQm` / i HBp2bX aQK2iBK2
biQ`2/ b }H2 QM vQm` +QKTmi2`- 2X;X +bp- 1t+2H- aSaa- Q` bQK
q?2M i?2 / i Bb QM vQm` +QKTmi2`- r2 b v i?2 / i Bb biQ`2/

. i + M HbQ #2 biQ`2/ 2ti2`M HHv QM i?2 AMi2`M2i- BM T +F ;2- Q` Q#
i?`Qm;? Qi?2` bQm`+2bX 6Q` 2t KTH2- bQK2 _ T +F ;2b +QMi BM / i b2ib
Mv+7HB;?Tb+RF ;2 +QMi BMb BM7Q`K iBQM QM ~B;?ib i? i /2T `i2/ Lu* BM
kyRjX

```
O MQi `mM
O BMbi HHXT +F ;2bU]Mv+7HB;?ibRj]V
HB#` `v UMv+7HB;?ibRjV
/ i U7HB;?ibV
M K2b U7HB;?ibV
O= (R) ]v2 `]      ]KQMi?]     ]/ v]        ]/2TniBK2]
O= (8) ]b+?2/n/2TniBK2] ]/2Tn/2H v]  ]`vniBK2]   ]b+?2/n ``niBK2]
O= (N) ] ``n/2H v]]+ ``B2`]     ]7HB;?i]     ]i BHMmK]
O= (Rj) ]Q`B;BM]  ]/2bi]       ] B`niBK2]   ]/Bbi M+2]
O= (Rd) ]?Qm`]    ]KBMmi2]    ]iBK2n?Qm`]
`K U7HB;?ibV
O
```

G i2` BM i?Bb +Qm`b2- r2öHH /Bb+mbb ?Qr iQ Q#i BM / i 7`QK r2# SAb M/
6Q` MQr- i?2 `2bi Q7 i?2 mMBi /Bb+mbb2b HQ+ H XHbv i? i Bb biQ`2/

NXkXk . i aiQ` ;2

A/2 HHv- vQm` / i b?QmH/ #2 biQ`2/ BM +2`i BM bpH2 7Q`K iX A `2+QKK2
U+QKK b2T ` i2/ p Hm2V }H2- r?B+? 7Q`K ib bT`2 /b?22i U`2+i M;mH `V
TH BM@i2ti 7Q`K iX `2 TH BM@i2ti- M/ + M #2 `2 / BMiQ HKQbi Mv
bi iBbiB+ H bQ7ir `2 T`Q;` K- BM+Hm/BM; _X h`v iQ pQB/ 1t+2H }H2b B7 v

>2`2 `2 bQK2 Qi?2` iBTb,

- q?2M rQ`FBM; rBi? bT`2 /b?22ib- i?2 }`bi `Qr Bb mbm HHv `2b2`p2/
  i?2 ?2 /2`- r?BH2 i?2 }`bi +QHmKM Bb mb2/ iQ B/2MiB7v i?2 b KTHBM
  UmMB[m2 B/2MiB}2`V
- pQB/ }H2 M K2b M/ p `B #H2 M K2b rBi? #H MF bT +2bX h?Bb + M +
  2``Q`b r?2M `2 /BM; BM / i X
- A7 vQm r Mi iQ +QM+ i2M i2 rQ`/b BM p `B #H2- 2Bi?2` M iQ rQ`/b
  BMbi2 / Q7 bT +2c
- a?Q`i M K2b `2 T`272`2/ Qp2` HQM;2` M K2bc
- h`v iQ pQB/ mbBM; M K2b i? i +QMi BM bvK#QHb bm+? b
  V- @O-\---I-=-f-%-$-(-)-&- M/c
- K F2 bm`2 i? i Mv KBbbBM; p Hm2b BM vQm` / i b2i `2 BM/B+ i2/ rBi?
  Q` #H MF }2H/b U/QMöi mb2 NN Q` ddVXX

+FMQrH2/;2K2Mib

h?Bb T ;2 Bb BM T `i /2`Bp2/ 7`QK i?2 7QHHQrBM; bQm`+2b, RX_ 7Q` . i a+B2M+2Mb2/ *mM/i`B p2 *QKKQMb ii`B#miBQM@ LQM*QKK2`+B H@LQ.2`Bpb jXy

kX 2MixFQr- J ii?2r M/ C2bb2 JX a? TB`QX kyR9X *Q/2 M/ . i 7 aQ+B H a+B2M+2b, S` +iBiBQM2`öb :mB/2X

NXj AKTQ`iBM; M/ 1tTQ`iBM;

NXjXR AKTQ`iBM; . i

6BM/ S i?b 6B`bi

AM Q`/2` iQ BKTQ`i UQ` `2 /V / i BMiQ _- vQm }`bi ? p2 iQ FMQr r?2 ?Qr iQ }M/ BiX

6B`bi- `2K2K#2` i? i vQmöHH M22/ iQ FMQr r?2 ?Qr iQ }M/ B`2+iQ`v vQm FMQr r?2`2 _ Bb HQQFBM; 7Q` }H2bX A7 vQmö`2 mbBM; _ S`Q /B`2+iQ`v rBHH #2 i?2 iQT@H2p2H /B`2+iQ`v Q7 i?2 T`QD2+iX

a2+QM/- vQmöHH M22/ iQ FMQr r?2`2 i?2 / i }H2 Bb- `2H iBp2 iQ /B`2+iQ`vX A7 Biöb .bi f`2rf 7QH/2`- i?2 `2H iBp2 T i? iQ vQm` }H2 rBHH.#2f` rf7BH2@M K2X+bp

_2 /BM; h #mH ` . i

h?2 rQ`F?Q`b2 7Q` `2 /BM; BMiQ /Xii#H2`BM;+bHHQrb Mv b2T ` iQ` U*ao- i #@/2HBK2Bi2/ Bp2b VXTV2+B H2+/X2 @F2UV 7Q` *ao }H2bX

h?2 # bB+ 7Q`KmH Bb,

```
O " bB+ *ao `2 /, AKTQ`i / i rBi? ?2 /2` `Qr- p Hm2b b2T ` i2/ #
Kv/ i b2i l`@ /X+bp UT BH2 4]-bi`BM;b b6 +iQ`b 4
```

>2`2öb T` +iB+ H 2t KTH2- mbBM; i?2 TQHBivoA / i b2i,

```
O BKTQ`i TQHBiv
TQHBiv l`@ /X+bp UT BH2 4]-bi`BM;b b6 +iQ`b 4 6 VV
TQHBiv( R,8- R,8)
O=     +v2 ` ++Q/2 b+Q/2     +QmMi`v v2 `
O= R dyyR3yy   dyy    6:  7;? MBbi M R3yy
O= k dyyR3yR   dyy    6:  7;? MBbi M R3yR
O= j dyyR3yk   dyy    6:  7;? MBbi M R3yk
O= 9 dyyR3yj   dyy    6:  7;? MBbi M R3yj
O= 8 dyyR3y9   dyy    6:  7;? MBbi M R3y9
```

NXjX AJSP_hAL: L.1sSP_hAL: Ry8

q2 mb2i`BM;b b6 +iQ`bM4 Q6`/2` iQ i`2 i i2ti +QHmKMb b +? ` +i2`
p2+iQ`b- MQi b 7 +iQ`bX A7 r2 /QMöi b2i i?Bb- i?2 /27 mHi Bb i? i HH MQM
+QHmKMb rBHH #2 2M+Q/2/ b 7 +iQ`bX h?Bb #2? pBQ` mbm HHv K F2b
M/ Bb /m2 iQ ?BbiQ`B+ H `2 bQMbX i QM2 TQBMi BM iBK2- 7 +iQ`b r2`2
+? ` +i2` p2+iQ`b- bXQ _ #Hb2 2Vi?2 /27 mHi iQ `2 / BM i2ti b 7 +iQ`bX

`2 /Xi #H2UV MmK#2` Q7 Qi?2` QTiBQMb,

O 6Q` BKTQ`iBM; i #mH ` / i rBi? K tBKmK +mbiQKBx2 #BHBiv
Kv/ i b2i I@ /Xi #H2UV2BH2 42 /2-4b2T-4[mQi2 42+-47BH-H4i`BM;b b6 +iQ`b4

_2 /BM; 1t+2H 6BH2b

.QMöi mb2 JB+`QbQ7i 1t+2H }H2b UXtHb Q` XtHbtVX "mi B7 vQm Kmbi,

O J F2 bm`2 vQm ? p2 BMbi HH2/ i?2 iB/vp2`b2 bmBi2 UQMHv M2+2bb `
O BMbi HHXT +F ;2bU]iB/vp2`b2]V O LQi _mM

O GQ / i?2]`2 /tH] T +F ;2 UM2+2bb `v 2p2`v M2r _ b2bbBQMV
HB#` Uv2 /tHV

`2 /n2t+2HUV b #QiH?b M/tHbt}H2b M/ /2i2+ib i?2 7Q`K i 7`QK i?2
2ti2MbBQMX

O " bB+ + HH
Kv/ i b2i I@ /n2t+2HUT i? 4- b?22i 4V

>2`2öb `2 H 2t KTH2,

O 1t KTH2 rBi? XtHbt UbBM;H2 b?22iV
B` I@2 /n2t+2HUT i f B`HBM2nbK H-tX?2HbitRV
B` (R 8- R 8)
O= O iB##H2, 8 t 8
O= u2 ` JQMi? . vQ7JQMi? . vP7q22F .2ThBK2
O= I/#H= I/#H=I/#H= I/#H= I+?`=
O= R kyy8 RR kk k Rdyy
O= k kyy3 R jR 9 kkRe
O= j kyy8 d Rd d Ny8
O= 9 kyy3 N kj k 38N
O= 8 kyy8 j 8 e 3kd

_2 /BM; ai i UX/i V 6BH2b

h?2`2 `2K Mvr vbXiQ }H2 /b BMiQ _X A `2+QK K2M/ mbBM;
Bi Bb T `i QE7/iv?p2`b2X

```
HB#` U? p2MV
 B`X/i I@2 /n/iU]/ i f B`HBM2nbK MHX/i ]
 B`(R8- R8)
O= O   iB##H2, 8 t 8
O=   u2 ` JQMi?. vQ7JQMi?. vP7q22F .2ThBK2
O=   I/#H= I/#H=I/#H=    I/#H= I+?`=
O= R  kyy8   RR  kk      k Rdyy
O= k  kyy3    R   jR      9 kkRe
O= j  kyy8    d   Rd      d Ny8
O= 9  kyy3    N   kj      k 38N
O= 8  kyy8    j    8      e 3kd
```

6Q` _2 HHv "B; . i

A7 vQm ? p2 `2 HHv #B; / i B2 #X+bp irBi? HH #2 iQQ bHQrX AM i?2b2 + b2b- +?2 Qmi i?2 7QHHQrBM; QTiBQMb,

RV 2 /n+bpBUMi?2 /`T +F ;2 Bb 7 bi2`- KQ`2 ?2HT7mH /`QT@BM `2 K2Mi 7Q`/X+bpiU V TH vb r2HH rBi? iB/vp2`b2 T +F ;2b U/Bb+m BM 7mim`2 H2bbQMbVX

kV i?2/ i Xi #H2 +F ;2 Bb ;`2 i 7Q` `2 /BM; M/ K MBTmH iBM; H / i b2ib UQ`/2`b Q7 ;B; #vi2b Q` Ryb Q7 ;B; #vi2bV

NXjXk 1tTQ`iBM; . i

uQm b?QmH/ M2p2` ;Q 7`QK ` r / i iQ `2bmHib BM QM2 b+`BTiX h r Mi iQ BKTQ`i ` r / i - +H2 M Bi- M/ i?2M 2tTQ`i i? i +H2 M2/ / i vQm` +QKTmi2`X h? i +H2 M2/ / i b2i rBHH i?2M #2 BKTQ`i2/ BMiQ 7Q` M HvbBb- BM KQ/mH ` 7 b?BQMX

hQ 2tTQ`i UQ` r`Bi2V / i 7`QK _ QMiQ vQm` +QKTmi2`- vQm + M +`2 X+bp}H2b- Q` 2tTQ`i K Mv / i Q#D2+ib iQ #BM2 `v}H2b- Q` #BM2`v }H2b- Q` Q#D2+iQ XM

q`BiBM; +bpaT`2 /b?22i

hQ 2tTQ`i M BM/BpB/m H / i 7` K2 b r`Bi2X+bpU X2+bp 2/Bi2- mb2

```
O " bB+ + HH
r`Bi2X+bpU4 - 7BH2-4`QrXM K2b+4QHXM K2b 4V
```

G2iöb r`Bi2i?2 b2i b +bpX

```
O " bB+ + HH
r`Bi2X+bpUB`-]/ i f B`HBM2b-X+bpX]-`QrXM K2b 4 6V
```

S +F ;BM;.i BMiQ X_.i

aQK2iBK2b- Biöb ?2HT7mH iQ r`Bi2 b2p2` H /i 7` K2b iQM+2- iQ #2 mb2 M HvbBbX hQ /Q bQ- mb2 i?2 7mM+iBQM iQ +`2 i2 QM2 }H2 +QMi BMBM; K Mv _/i Q#D2+ibX

```
O " bB+ + HH
b p2U2XXX7-BH2 V4
```

>2`2öb ?Qr r2 + M r`Bi2 #Qi? Q#D2+ib iQ QM2 }H2X

```
b p2U2 B`- TQHB+v- 7BH2 ]/ i f/ i b2ibX_. i]V
```

q2 + M i?2M `2 /i?2b2 / i b2ib # +F BMiQ _ mbBM; HQ /UVX

```
O +H2 ` 2MpB`QMK2Mi
`KUHBbiUVV

O HQ / / i b2ib
HQ /U]/ i f/ i b2ibX_. i]V
```

NX9 1tTHQ`BM; . i

NX9XR h?2 : TKBM/2` . i b2i

h?Bb H2bbQM /Bb+mbb2b ?Qr iQ T2`7Q`K # bB+ 2tTHQ` iQ`v / i M HvbBb

6Q` i?Bb mMBi- r2öHH #2 rQ`FBM; rBi? i?2 ó: TKBM/2`ô / i b2i- r?B+? B 2t+2`Ti Q7 i?2 / i p BH #H2 i : TKBM/2`XQ`;X 6Q` 2 +? Q7 R9k +QmMi`B / i T`QpB/2b p Hm2b 7Q` HB72 2tT2+i M+v- :.S T2` + TBi - M/ TQTmH iBQ }p2 v2 `b- 7`QK RN8k iQ kyydX

```
; T I@ 2 /X+bpU]/ i f; TKBM/2`X+bp]- bi`BM;b b6 +iQ`b 4 6V
```

NX9Xk ai`m+im`2 M/ .BK2MbBQMb

h?2 }`bi i?BM;b r2 r Mi iQ FMQr #Qmi / i b2i `2 Bib /BK2MbBQMb M/ # bi`m+im`2X 6Q` BMbi M+2- r2 + M HQQF i i?2 MmK#2` Q7 `Qrb M/ +QHmK

```
O ;2i MmK#2` Q7 `Qrb M/ +QHmKMb,
/BKU; TV
O= (R) Rdy9 e
```

q2 KB;?i HbQ r Mi iQ b22 i?2 M K2b Q7 i?2 +QHmKMb,

```
O b22 +QHmKM M K2b
M K2b U; TV
O= (R) ]+QmMi`v] ]v2]T]QT] ]+QMiBM2Mi] ]HB721tT] ];/TS2
```

h?2 bi`7mM+iBQM Bb ?2HT7mH iQ b22 M Qp2`pB2r Q7 i?2 / i öb bi`

```
O b22 bi`m+im`2 Q7 / i
bi`U; TV
O= ^/ i X7` K2^, Rdy9 Q#bX Q7 e p `B #H2b,
O= 0 +QmMi`v , +?` ] 7;? MBbi M] ] 7;? MBbi M] ] 7;? MBbi M]
O= 0 v2 ` , BMi RN8k RN8d RNek RNed RNdk RNdd RN3k RN
O= 0 TQT , MmK 39k8jjj Nk9yNj9 Rykedy3j RR8jdNee RjydN9ey
O= 0 +QMiBM2Mi, +?` ] bB ] ] bB ] ] bB ] ] bB ] XXX
O= 0 HB721tT , MmK k3X3 jyXj jk j9 jeXR XXX
O= 0 ;/TS2`+ T, MmK ddN 3kR 38j 3je d9y XXX
```

6BM HHv- A 2M+Qm` ;2 vQm iQ +im HHv T?2 F7mi M?2 iBQMBib2H7X h
/BbTH vb i?2 }`bi e `Qrb Q7 Mv / i 7` K2X

```
?2 U; TV
O= +QmMi`v v2 `TQT +QMiBM2Mi HB721tT ;/TS2`+ T
O= R 7;? MBbi M RN8k 39k8jjj bB k3X3 ddN
O= k 7;? MBbi M RN8d Nk9yNj9 bB jyXj 3kR
O= j 7;? MBbi M RNek Rykedy3j bB jkXy 38j
O= 9 7;? MBbi M RNed RR8jdNee bB j9Xy 3je
O= 8 7;? MBbi M RNdk RjydN9ey bB jeXR d9y
O= e 7;? MBbi M RNdd R93k3jdk bB j3X9 d3e
```

NX9Xj *QKKQM Hi2` iBQMb

h?2`2 `2 bQK2 p2`v +QKKQM Hi2` iBQMb `2b2 `+?2`b K F2 QM i?2
BM;+QHmKM M K2b- bbB;MBM; L p Hm2b- M/ +? M;BM; +QHmKM

LQi2 i? i r2 rBHH +Qp2` ?Qr iQ T2`7Q`K i?2b2 Hi2` iBQMb mbBM;
H i2` BM i?2 +Qm`b2X >Qr2p2`- i?2b2 HBM2b `2 p2`v +QKKQM- bQ
?Qr i?2v rQ`F,

RX *? M;2 +QHmKM M K2b

```
M K2b U; TV
O= (R) ]+QmMi`v] ]v2]T]QT] ]+QMiBM2Mi] ]HB721tT] ];/TS2
M K2b U; TV I@ +U]+QmMi`v] ]v2 `] ]TQT] ]+QMiBM2Mi] ]HB72 ]T]
bi`U; TV
O= ^/ i X7` K2^, Rdy9 Q#bX Q7 e p `B #H2b,
O= 0 +QmMi`v , +?` ] 7;? MBbi M] ] 7;? MBbi M] ] 7;? MBbi M
O= 0 v2 ` , BMi RN8k RN8d RNek RNed RNdk RNdd RN3k RN3
O= 0 TQT , MmK 39k8jjj Nk9yNj9 Rykedy3j RR8jdNee RjydN9e
```

```
O= 0 +QMiBM2Mi , +?` ] bB ] ] bB ] ] bB ] ] bB ] XXX
O= 0 HB72X2tT , MmK k3X3 jyXj jk j9 jeXR XXX
O= 0 ;/TXT2`+ T, MmK ddN 3kR 38j 3je d9y XXX
```

kX*? M;2 bQK2 p Hm2b iQ

```
; T0HB72X2tT0(H;BT72X2tyT) I@L
```

jX*Q2`+2 +QHmKMb iQ bT2+B}+ ivT2X 6Q` BMbiM+2- H2iöb +? M;2 7`QK +? ` +i2` iQ 7 +iQ`X

```
bmKKU`;v0T+QMiBM2MiV
O= G2M;i? *H bbJQ/2
O= Rdy9 +? ` +i2` +? ` +i2`
; T0+QMiBM2MbX7@+iQ`; T0+QMiBM2MiV
bmKKU`;v0T+QMiBM2MiV
O= 7`B+ K2`B+ b bB 1m`QT2 P+2 MB
O= ek9 jyy jNe jey k9
```

NX9X9 amKK `v bi iBbiB+b

q2 + M ;2i [mB+F bmKK `v bi iBbiB+b 7Q` bbBM;; i?2 2MiB`2 / i 7` K2 rBHH bmKK `Bx2 HH +QHmKMb,

```
bmKKU`;vTV
O= +QmMi`v v2 ` TQT +QMiBM2Mi
O= G2M;i?,Rdy9 JBMX ,RN8k JBMX ,eXyy2Yy9 7`B+ ,ek9
O= *H bb ,+? ` +i2` Rbi ZmX,RNee Rbi ZmX,kXdN2Yye K2`B+ b,jyy
O= JQ/2 ,+? ` +i2` J2/B M ,RN3y J2/B M ,dXyk2Yye bB ,jNe
O= J2 M ,RN3y J2 M ,kXNe2Yyd 1m`QT2 ,jey
O= j`/ ZmX,RNNj j`/ ZmX,RXNe2Yyd P+2 MB , k9
O= J tX ,kyyd J tX ,RXjk2YyN
O= HB72X2tT;/TXT2`+ T
O= JBMX ,kjXe JBMX , k9R
O= Rbi ZmX,93Xk Rbi ZmX, Rkyk
O= J2/B M ,eyXd J2/B M , j8jk
O= J2 M ,8NX8 J2 M , dkR8
O= j`/ ZmX,dyX3 j`/ ZmX, Njk8
O= J tX ,3kXe J tX ,RRj8kj
```

S bbBM; +QHmKM rBi? bmKK `Bx2 i? i T `iB+mH ` +QHmKM,

```
bmKKU`;v0Tv2 `V
O= JBMX Rbi ZmX J2/B M J2 M j`/ ZmX J tX
O= RN8k RNee RN3y RN3y RNNj kyyd
```

aQK2iBK2b r2 M22/ iQ /Q bQK2 # bB+ +?2+FBM; 7Q` i?2 MmK#2` Q7 Q#b2`

Q` ivT2b Q7 Q#b2`p iBQMb BM Qm` / i b2iX hQ #2 KQ`2 [mB+FHv M2 iQ 2bi#HBb Bb Qm` 7`B2M/X

G2iöb HQQF i i?2 MmK#2` Q7 Q#b2`p iBQMb }`bi #v `2;BQM- M/ i?2 M/ v2 `X

```
i #H2; U+QMiBM2MiV
O=
O=    7`B+    K2`B+ b      bB    1m`QT2  P+2 MB
O=    ek9    jyy    jNe    jey     k9

i #H2; U+QMiBM2Mi- v2 `V
O=
O=         RN8k RN8d RNek RNed RNdk RNdd RN3k RN3d RNNk
O=    7`B+      8k   8k   8k   8k   8k   8k   8k   8k   8k   8k   8k
O=    K2`B+ b   k8   k8   k8   k8   k8   k8   k8   k8   k8   k8   k8
O=    bB     jj   jj   jj   jj   jj   jj   jj   jj   jj   jj   jj   jj
O=   1m`QT2     jy   jy   jy   jy   jy   jy   jy   jy   jy   jy   jy   jy
O=   P+2 MB      k    k    k    k    k    k    k    k    k    k    k    k
```

q2 + M 2p2M /BpB/2 #v i?2 iQi H MmK#2` Q7 `Qrb iQ ;2i T`QTQ`iBQ

```
i #H2; U+QMiBM2MiV f MQ`Qr; TV
O=
O=    7`B+    K2`B+ b      bB    1m`QT2  P+2 MB
O=   yXjeek   yXRdeR   yXkjk9   yXkRRj   yXyR9R
i #H2; U+QMiBM2MiV f MQ`Qr; TV Ryy
O=
O=    7`B+    K2`B+ b      bB    1m`QT2  P+2 MB
O=    jeXek    RdXeR    kjXk9    kRXRj    RX9R
```

NX9X8 _2pB2r Q7 am#b2iiBM;

q2 H2 `M2/ #Qmi bm#b2iiBM; BM i?2 T`2pBQmb H2bbQMX G2iöb ?2`2,

```
O 1ti` +i 7B`bi Ry `Qrb
; T(R,Ry)
O=      +QmMi`v v2 `TQT +QMiBM2Mi HB72XtT ;/TXT2`+ T
O= R   7;? MBbi M  RN8k  39k8jjj    k3X3    ddN
O= k   7;? MBbi M  RN8d  Nk9yBNj9   jyXj     3kR
O= j   7;? MBbi M  RNek  RykeddBy3j  jkXy    38j
O= 9   7;? MBbi M  RNed  RR8jdBNee  j9Xy    3je
O= 8   7;? MBbi M  RNdk  RjydNB9ey  jeXR    d9y
O= e   7;? MBbi M  RNdd  R93b3Bjdk  j3X9    d3e
O= d   7;? MBbi M  RN3k  Rk33BR3Re  jNXN    Nd3
O= 3   7;? MBbi M  RN3d  Rj3edBN8d  9yX3    38k
```

```
O= N   7;? MBbi M RNNk RejRRNkR 9RXd   e9N
O= Ry  7;? MBbi M RNNd kkkkd9R8 9RX3   ej8

O 1ti` +i +QmMiv v2 ` 7Q` 7B`bi Ry `Qrb
; T(R Ry +U]+QmMi`v]2 `V)
O=     +QmMi`v v2 `
O= R   7;? MBbi M RN8k
O= k   7;? MBbi M RN8d
O= j   7;? MBbi M RNek
O= 9   7;? MBbi M RNed
O= 8   7;? MBbi M RNdk
O= e   7;? MBbi M RNdd
O= d   7;? MBbi M RN3k
O= 3   7;? MBbi M RN3d
O= N   7;? MBbi M RNNk
O= Ry  7;? MBbi M RNNd

O 1ti` +i Q#b2`p iBQMb BM  7`B+
 7`B+  I;@T(; 0T+QMiBM2Mi]7`B+-]
O 6BM/  p2` ;2 HB72 2tT2+i M+v 7Q` Q#b2`p iBQMb BM  7`B+
K2 UM; 0HB72X2tT(+QMiBM2Mi]7`B+)V
O= (R) 93XN
K2 UM 7`B+0HB72X2tTV
O= (R) 93XN
```

NX9Xe " bB+ SHQiiBM;

q2öHH ;Q BMiQ THQiiBM; BM ;`2 i2` /2i BH bQQM- #mi H2iöb iQm+? QM i ;` T?bX 6B`bi- b+ ii2`THQi,

```
THQi(UM; 0HB72X2tT;T 0;/TXT2`+ TV
```

?BbU; THB72X2#T2 FbR4yyV

NX9Xd *? HH2M;2b

*? HH2M;2 RX

_2 /i?2QHBiv b2iX

*? HH2M;2 kX

_2TQ`ii?2 MmK#2` M/M K2b Q7 2 +?p `B #H2 BM i?2 /i b2iX

*? HH2M;2 jX

1ti` +i i?2 8i?`Qr 7`QK i?2 TQHBiv /i b2iX

*? HH2M;2 9X

1ti` +i i?2 H bi `Qr 7`QK i?2 TQHBiv /i b2iX

*? HH2M;2 8X

*QmMi i?2 T2`+2Mi ;2 Q7 Q#b2`p iBQMb; Bi?2`i?2 TQHBiv ?M 3 Q7M i?2 ; TKBM/2` /i b2iX

U>BMi, B7 mbBM; bmKUV-`2 /i?2 ?2HT }H2XXV

*? HH2M;2 eX

a2i HH Q7 i?2 p2Km2Mb Q7 +QHmKMb H2bb i? M @Ry iQ L X

U>BMi, uQm b?QmH/ }`bi +QMbi`m+i M/ rQ`F QM i?2 +QTv bQ i? i i?2 Q`B;BM H / i b2i Bb mM+? M;2/ UQ` Dmbi `2 /i?2 / i BMiQ _ ; BM 7i2

iQ ;2i +H2 M +QTvV

*? Ti2` Ry

. i h` Mb7Q`K iBQM

RyXR AMi`Q/m+iBQM

RyXRXR P/vp2`b2

Ai Bb Q7i2M b B/ i? i 3yW Q7 / i M HvbBb Bb bT2Mi QM i?2 T`Q+2bb Q
+H2 MBM; M/ T`2T `BM; i?2 / i X U. bm M/ CQ?MbQM- kyyjV

6Q` KQbi TTHB2/ `2b2 `+?2`b- / i T`2T ` iBQM mbm HHv BMpQHp2b j K B

RX h` Mb7Q`KBM; 7` K2b- 2X;X }Hi2`BM;- bmKK `BxBM;- M/ +QM/m+iBM;
+ H+mH iBQMb +`Qbb;`QmTbX
kX hB/vBM; i BMiQ i?2 TT`QT`B i2 7Q`K iX
jXJ2`;BM; Q` HBMFBM; b2p2` H / i b2ib iQ +`2 i2 #B;;2` / i b2iX

h?2 iB/vp2`b2 bmBi2 Q7 T +F ;2b /2bB;M2/ bT2+B}+ HHv iQ ?2HT rBi? i?
bi2TbX h?2b2 `2 #v MQ K2 Mb i?2 QMHv T +F ;2b Qmi i?2`2 7Q` / i r` M;
#mi i?2v `2 BM+`2 bBM;Hv TQTmH ` 7Q` i?2B` `2 / #H2- bi` B;?i7Q`r `/ bv
b2MbB#H2 /27 mHi #2? pBQ`bX

AM i?Bb +? Ti2`- r2ö`2 ;QBM; iQ 7Q+mb QM F2v 7mM+iBQMb i?i b2 i?2
i` Mb7Q`K iBQM i bFbX

RyXRXk : TKBM/2`

6Q` i?Bb mMBi- r2öHH #2 rQ`FBM; rBi? i?2 ó: TKBM/2`ô / i b2i ; BMX

; T I@2 /X+U]p i f; TKBM/2`@6Bp2u2 `-bi `X+Mp] b6 +iQ`b_lY4
F #H22 U; TVV

+QmMi`v

v2 `

TQT

+QMiBM2Mi

HB721tT

;/TS2`+ T

7;? MBbi M

RN8k

39k8jjj

bB

k3X3

ddN

7;? MBbi M

RN8d

Nk9yNj9

bB

jyXj

3kR

7;? MBbi M

RNek

Rykedy3j

bB

jkXy

38j

7;? MBbi M

RNed

RR8jdNee

bB

j9Xy

3je

7;? MBbi M

RNdk

RjydN9ey

bB

jeXR

d9y

7;? MBbi M

RNdd

R933yjdk

bB

j3X9

d3e

RyXRXj q/?TvHv`

aQ 7 `- vQmöp2 b22M i?2 # bB+b Q7 K MBTmH iBM; / i 7` K2b- 2X;X bm#b
bB+ + H+mH iBQMbX 6Q` BMbi M+2- r2 + M mb2 # b2 _ 7mM+iBQMb iQ +
K `v bi iBbiB+b +`Qbb ;`QmTb Q7 Q#b2`p iBQMb- 2X;X- i?2 K2 M :.S T2`
rBi?BM 2 +? `2;BQM,

```
K2 UM; T(;0+QMiBM2Mi4]7`B+-]];/TS2`+)V]
O= (R) kRN9
K2 UM; T(;0+QMiBM2Mi4]K2`B+-b]];/TS2`+)V]
O= (R) dRje
K2 UM; T(;0+QMiBM2Mi4]bB -]];/TS2`+)V]
O= (R) dNyk
```

"mii?Bb BbMöiB/2 H #2+ mb2 Bi BMpQHp2b 7 B`#Bi Q7`2T2iBiBQMX _2T
rBHH +Qbi vQm iBK2- #Qi? MQr M/ H i2`- M/ TQi2MiB HHv BMi`Q/m+2 b
#m;bX

Gm+FBHv- i?2 +F ;2 T`QpB/2b MmK#2` Q7 p2`v mb27mH 7mM+iBQMb 7Q` K
mH iBM; / i 7` K2bX h?2b2 7mM+iBQMb rBHH b p2 vQm iBK2 #v `2/m+BM;
b M //2/ #QMmb- vQm KB;?i 2p2M]HK/ i?2 bB2` iQ `2 /X

>2`2- r2ö`2 ;QBM; iQ +Qp2` e Q7 i?2 KQbi mb27mH +QKK M/b 7Q` K MBTmH
HbQ +Qp2` i?2 `2bi `B+i2/ bB?mHi`2BM; 7mM+iBQMb

RX b2H2+iUV
kX 7BHi2`UV
jX ;`QmTn#vUV
9X bmKK `Bx2UV
8X Kmi i2UV
eX `` M;2UV

A7 vQm ? p2 ? p2 MQi BMbi HH2/ i?Bb T +F ;2 2 `HB2`- TH2 b2 /Q

```
O MQi `mM
O BMbi HHXT +F ;2bU^/THv`^V
```

LQr H2iöb HQ / i?2 T +F ;2,

```
HB#` Uv/THv`V
```

RyXk /THv6mM+iBQMb

RyXkXR a2H2+i *QHmKMb+ r Bi? mKH2+ i Bi?

AK ;BM2 i? i r2 Dmbi `2+2Bp2/ i?2 ; TKBM/2` / i b2i- #mi `2 QMH BM 72r p `B #H2b BM BiX q2 H2 `M i? i r2 ?2 iQ F22T QMHv i?2 p `B #H2b r2 b2H2+iX

```
v2 `n+QmMi`vn;/TH@+ T- v2 `- +QmMi`v- ;/TS2`+ TV
F #H2 Uv2 `n+QmMi`vn;/TVV
```

v2 `

+QmMi`v

;/TS2`+ T

RN8k

7;? MBbi M

ddN

RN8d

7;? MBbi M

3kR

RNek

7;? MBbi M

38j

RNed

7;? MBbi M

3je

RNdk

7;? MBbi M

d9y

RNdd

7;? MBbi M

d3e

```
FMB,i,`BM+Hm/2n;`UT?iB?+4]BK;f/THv`@7B,VRXTM;]
```

A7 r2 QT2M2 mT+QmMi`v n2;/TH b22 i? iBi QMHv +QMi BMb i?2 v2 `- +QmM@
i`v M/ ;/TS2`+ TX h?Bb Bb 2[mBp H2Mi iQ i?2 # b2 _ bm#b2iiBM; 7mM+iBQ

```
v2 `n+QmMi`vn;/Tn;# Tb(2+U]v@2 `-] ]+QmMi`v;/TS2`+T])
F #HU2 UUv2 `n+QmMi`vn;/TVV
```

v2 `

+QmMi`v

;/TS2`+ T

RN8k

7;? MBbi M

ddN

RN8d

7;? MBbi M

Rky *> Sh1_ RyX .h h_ La6P_J hAPL

3kR

RNek

7;? MBbi M

38j

RNed

7;? MBbi M

3je

RNdk

7;? MBbi M

d9y

RNdd

7;? MBbi M

d3e

"mi- br2 rBH/FHbvK2-F2b 7Q` Km+? KQ`2 `2 /B#H2- 2{+B2Mi +Q/2 #2
Q7 BTiBbTQ2T2` iQ`X

RyXkXk h?2 SBT2

```
FMB,i,`BM+Hm/2n;`UT?iB?+4]BK;fTBT2XVDT;]
```

#Qp2-r2 mb2/r? iöb + HH2/õMQ`K Hö;` KK/TH#?#mB2?2 bi`2M;i?b Q7
BM +QK#BMBM; b2p2` HT7mTX2aBBQMQ+2bi?mb2bBMT;2b ;` KK ` Bb mMHBF2
Mvi?BM; r2öp2 b22M BM _ #27Q`2- H2iöb `2T2 i r? i r2öp2 /QM2 #Qp2 mbB

AM ivTB+ H # b2 _ +Q/2- bBKTH2 QT2` iBQM KB;?i #2 r`Bii2M HBF2,

O LPh `mM
+mT+ F2b # I@2T Q mK BU UBM;`2/B2MibVVV

+QKTmi2` ? b MQ i`Qm#H2 mM/2`bi M/BM;i?Bb M/vQm` +mT+ F2b rBHH
Dmbi }M2- #mi T2`bQM ? b iQ `2 / `B;?i iQ H27i iQ mM/2`bi M/ i?2 Q`/2
QT2` iBQMb @ i?2 QTTQbBi2 Q7 ?Qr KQbi r2bi2`M H M;m ;2b `2 `2 / @ K
? `/2` iQ mM/2`bi M/ r? i Bb #2BM; /QM25

hQ #2 KQ`2 `2 / #H2 rBi?Qmi TBT2b- r2 KB;?i #`2 F mT i?Bb +Q/2 BMiQ BM
/B i2 Q#D2+ib,

OO LPh `mM
ii2` I@ BU UBM;`2/B2MibV
Km77BMniBM/ I@ ii2`V
+mT+ F2b # I@2Km77BMniBMV

"mi- i?Bb + M +Hmii2` Qm` 2MpB`QMK2Mi rBi? HQi Q7 p `B #H2b i? i `2
mb27mH iQ mbX SHmb- i?2b2 p `B #H2b `2 Q7i2M `2 M K2/ p2`v bBKB

U2X;X bi2T- bi2TR- bi2Tk V r?B+? + M H2 /iQ +QM7mbBQM M/ i?2
iQ@i` +F@/QrM #m;bX

1Mi2` i?2 SBT2

h?2 TBTK2F2b Bi 2 bB2` iQ `2 / +Q/2 #v H vBM; Qmi QT2` iBQMb 7`QK
2 +? HBM2 + M #2 `2 / HBF2 HBM2 Q7 `2+BT2 7Q` i?2 T2`72+i /

SBT2b i F2 i?2 BMTmi QM i?2 H27i #QH2 M/ T i?2b Bi BM b i?2 }`bi
`;mK2Mi iQ i?2 7mM+iBQM QM i?2 `B;?i bB/2X

qBi? TBT2b- Qm` +mT+ F2 2t KTH2 KB;?i #2 r`Bii2M HBF2,

OO LPh `mM
+mT+ F2b BM@`2/B2Mib W=W
KBt UV W=W
TQm` UV W=W
F2 UV

hBTb 7Q` SBTBM;

RX_2K2K#2` i? i vQm /QMöi bbB;M Mvi?BM; rBi?BM i?2 TBT2b
b?QmH/ MQi mb2 I@ BMbB/2 i?2 TBT2/ QT2` iBQMX PMHv mb2
MBM; Q7 vQm` +Q/2 B7 vQm r Mi iQ b p2 i?2 QmiTmiX

kX_2K2K#2` iQ // i?2 W=W i?2 2M/ Q7 2 +? HBM2 BMpQHp2/ BM i?
TBT2/ QT2` iBQMX ;QQ/ `mH2 Q7 i?mK#, bBM+2 _aim/BQ rBH
BM/2Mi HBM2b Q7 +Q/2 i? i `2 T `i Q7 TBT2/ QT2` iBQM- B7
BM/2Mi2/- Bi T`Q# #Hv ? bMöi #22M //2/ iQ i?2 TBT2X A7 vQm
BM TBT2/ QT2` iBQM- Hr vb +?2+F iQ K F2 bm`2 i?2 TBT2 Bb
vQm 2tT2+iX

jXAM _aim/BQ- i?2 ?QiF2v 7Q` i?2 TBT2 Bb *i`H Y a?B7i Y JX

b2H2+i SBT2 W=W

aBM+2 i?2 TBT2 ;` KK ` Bb mMHBF2 Mvi?BM; r2öp2 b22M BM _ #2
r? i r2 /B/ #Qp2 rBi? i?2 ; TKBM/2` / i b2i mbBM; TBT2b,

v2 `n+QmMi`vn;/T W@W 2H2+i U +v2 `- +QmMi`v- ;/TS2`+ TV

G2iöb r HF i?`Qm;? Bi bi2T #v bi2TX

6B`bi- r2 bmKKQM i?2 ; TKBM/2` / i 7` K2 M/ T bb Bi QM iQ i?2 K
mbBM; i?2 TBT2 W=W K#QH

h?2 b2+QM/ bi2T Bb i?2 7mM+iBQMX AM i?Bb + b2- r2 /QMöi bT2+B7v r?B+? / i Q#D2+i r2 mb2 BM i?2 2+HBTb2 r2öp2 TBT2/ Bi BM 7`QK i?2 T`2pBQmb HBM2X

6mM 6 ,+h ?2`2 Bb ;QQ/ +? M+2 vQm ? p2 2M+QmMi2`2/ TBT2b #27Q`2 BM b?2HHX AM _- TBT2b #2+QK2 b?2HH BMi2`T`2i2/ TBT2b #v i?2 b K25

RyXkXj 6BHi2` _Qr Bb i?

LQrH2iöb b vr2ö`2 QMHv BMi2`2bi2/ BM 7`B+bM2H2+iMi`B2bX q2 + M +QK# M/7BHi2Q b2H2+i QMHv i?2 Q#b2`p iBQMb r?B+? `2

```
v2 `n+QmMi`vn;/Tn 7`B+ W=W@
    7BHi2` +QMiBM2Mi 44 ]7`B+ ]W=W
    b2H2+i v2 `- +QmMi`v- ;/TS2`+ TV
```

b rBi? H bi iBK2- }`bi r2 T bb i?2 ; TKBM/2` /Hii2` UK2 iQ i?2 7mM+iBQM- i?2M r2 T bb i?2 }Hi2`2/ p2`bBQM Q7 i?2 ; TKBM/2` / i 7` K2 b2H2+i 7mM+iBQMX

hQ +H `B7v- #2+ mb2 7BHi2` 7mM+iBQMb bm#b2i i?2 / i 7` K2X h?2 /Bz2`2M+2 #2ir22M +ib +2`i BM +QH7mKMb- r?B+? Bb+2`i BM `QrbX

LQi2 i?2 Q`/2` Q7 QT2` iBQMb Bb p2`v BKTQ`i MiX AM i?Bb + b2X A7 r2 mb2/ }`bi- }Hi2` rQmH/ MQi #2 #H2 iQ }M/ i?2 p `B #H2 +QmMi`v +QmH/ ? p2 `2KQp2/ Bi BM i?2 T`2pBQmb bi2TX

RyXkX9 * H+mH i2 +`Qbb ;`QmTb bm# i?

+QKKQM i bF vQmöHH 2M+QmMi2` r?2M rQ`FBM; rBi? / i Bb `mMMBM; + H QM /Bz2`2Mi ;`QmTb rBi?BM i?2 / i X 6Q` BMbi M+2- r? i B7 r2 r Mi2/ iQ + H i?2 K2 M :.S T2` + TBi 7Q` 2 +? +QMiBM2Mi\

AM # b2 _- vQm rQmH/ ? p2 iQ 7mM+iBQM 7Q` 2 +? bm#b2i Q7 / i ,

```
K2 M U;/TS2`+ T(+QMiBM2Mi 44 ]7`B+ )V
O= (R) kRN9
K2 M U;/TS2`+ T(+QMiBM2Mi 44 ]K2`B+ )V]
O= (R) dRje
K2 M U;/TS2`+ T(+QMiBM2Mi 44 ]bB )V
O= (R) dNyk
K2 M U;/TS2`+ T(+QMiBM2Mi 44 ]1m`QT2 ]V
O= (R) R99eN
K2 M U;/TS2`+ T(+QMiBM2Mi 44 ]P+2 MB )V ]
O= (R) R3ekk
```

h? iöb HQi Q7 `2T2iBiBQM5 hQ K F2 K ii2`b rQ`b2- r? i B7 r2 r M
i?2b2 p Hm2b iQ Qm` Q`B;BM H / i 7` K2 b M2r +QHmKM\ q2 rQ
r`Bi2 bQK2i?BM; HBF2 i?Bb,

```
; T0K2 MX+QMiBM2MiX:.S I@
; T0K2 MX+QMiBM2M0+XQ.SiB;MP24]i7`B+)]I@K2 U; T;/TS2`+ T(0+QMiBM
; T0K2 MX+QMiBM2M0+XQ.SiB;MP24]iK2`B+) b]I@K2 U; T;/TS2`+ T(0+QMiB
; T0K2 MX+QMiBM2M0+XQ.SiB;MP24]ibB )]I@K2 U; T;/TS2`+ T(0+QMiBM24
; T0K2 MX+QMiBM2M0+XQ.SiB;MP24]1m`QT2 ]I@K2 U; T;/TS2`+ T(0+QMiBM
; T0K2 MX+QMiBM2M0+XQ.SiB;MP24]P+2 MB )]I@K2 U; T;/TS2`+ T(0+QMiB
```

uQm + M b22 ?Qr i?Bb + M ;2i T`2iiv i2/BQmb- 2bT2+B HHv B7 r2 r
KQ`2 +QKTHB+ i2/ Q` `2}M2/ bi iBbiB+bX q2 +QmH/ mb2 HQQTb Q
#mi i?2b2 + M #2 /B{+mHi- bHQr- M/ 2``Q`@T`QM2X

bTHBi@ TTHv@+QK#BM2

h?2 #bi` +i T`Q#H2K r2ö`2 2M+QmMi2`BM; ?2`2 Bb FMQr b óbTHB

```
FMB,i,`BM+Hm/2n;` UT?iB+4]BK;fbTHBi TTHvXTM;]
```

q2 r Mi iQTHQm` / i BMiQ ;`QmTb UBM i?Bb+ b2- +QMiBM2MibV-
+ H+mH iBQMb QM i?+iQ;K#BM2 i?2 b2 M2r iQ;2i?2` 7i2`r `/bX

Gm+FBHv-Qz2`b Km+? +H2 M2`- bi` B;?i@7Q`r `/ bQHmiBQM iQ i?

6B`bi- H2iöb `2KQp2 i?2 +QHmKM r2 Dmbi K /2X

```
; T I@ T W=W2H2U@K2 MX+QMiBM2MiX:.S V +QHmKM rBi? @
O P_
; T0K2 MX+QMiBM2MiX:.S I@
```

RyXkX9 ;XQmTn#v

q2öp2 H`2 /v b27BMH?2Q`U M ?2HT mb b2H2+i Q#b2`p iBQMb i? i K22i +2`i BM +`Bi2`B UBM Q`/2` iQ #2 Q`44 ]VX 3Q` 2 ?2HT7mH- ?Qr2p2`- Bb i?2 ;`QmTn#v UM+iBQM- r?B+? rBHH 2bb2MiB HHv mb2 2p2`v mMB[m2 +`Bi r2 +QmH/ ? p2 7BH2i2`XUV

;`QmT2/n+/7M #2 i?Qm;?i Q7 b i? 2`2 2 +? Bi2KH BMbB/2 / i X7` K2 +? +QMi BMb QMHv i?2 `Qrb i? i +Q``2bTQM/ iQ T `iB+mH `p 7Q+`QMiBM2Mi H2 bi BM i?2 2t KTH2 #Qp2VX

FMBi,`BM+Hm/2n;`UT ?iB+4]BK;f/THv`@7B;kXTM;]V

RyXkX8 amKK `Bx2 +`Qbb :`QmTb

;`QmTn#v UV Bib QrM Bb MQi T `iB+mH `Hv BMi2`2biBM;X Aiöb Km+? KQ`2 mb2/ BM +QMDmM+iBQM rBi? bmKK `Bx2UV

h?Bb rBHH HHQr mb iQ +`2 i2 M2r p `B #H2UbV #v TTHvBM; i` Mb7Q`K p `B #H2b BM 2 +? Q7 Qm` ;`QmTb U+QMiBM2Mi@bT2+B}+ / i 7` K2bVX

AM Qi?2` rQ`/b- m`QmTn#v UM+iBQM- r2 bTHBi Qm` Q`B;BM H / i 7` K2 BMiQ KmHiBTH2 TB2+2b- iQ r?B+? r2 i?2M TTHv bmKK `v 7mM+iBQMb U2X;X Q`b/UV rBi? bmKK `Bx2UV

h?2 QmiTmi Bb M2r / i 7` K2 `2/m+2/ BM bBx2- rBi? QM2 `Qr T2` ;`QmTX

```
;/Tn#v+QMiBM2MiT b W=W
  ;`QmTn#v U+QMiBM2MiV W=W
  bmKK `Bx2 UK2Mn;/TS2`+ T M;/TS2`+ TVV
F #H2 U;/Tn#v+QMiBM2MibVV
```

+QMiBM2Mi

K2 Mn;/TS2`+ T

R k e *> S h 1 _ R y X . h h _ L a 6 P _ J h A P L

7 ` B +

k R N 9

K 2 ` B + b

d R j e

b B

d N y k

1 m ` Q T 2

R 9 9 e N

P + 2 M B

R 3 e k k

```
F M B i , ` B M + H m / 2 n ; ` U T ?iB +4 ] B K ; f / T H v ` @ 7 B ; ] X T M ; ]
```

h? i H H Q r 2 / m b i Q + H + m H i 2 i? 2 K 2 M ; / T S 2 ` + T 7 Q ` 2 +? + Q M i B M

" m i B i ; 2 i b 2 p 2 M # 2 i i 2 ` j ? 2 7 T M + i B Q M b m b i Q ; ` Q m T # v K m H i B T H 2
p ` B # H 2 b X G 2 i ö b 2 ; ` Q m T # v i B M X 2 M i

```
; / T n # v + Q M i B M 2 M i b n # v v 2 W = ` W i @
    ; ` Q m T U # v + Q M i B M 2 M i - v 2 W = W ` V
    b m K K ` B x 2 M n ; / T S 2 ` + K 2 T M ; / T S 2 ` + T V V
F # H 2 U ; / T n # v + Q M i B M 2 M i b n # v v 2 ` V V
```

+ Q M i B M 2 M i

v 2 `

K2 Mn;/TS2`+ T

7`B+

RN8k

Rk8j

7`B+

RN8d

Rj38

7`B+

RNek

R8N3

7`B+

RNed

ky8y

7`B+

RNdk

kj9y

7`B+

RNdd

k83e

h? i Bb H`2 /v [mBi2 TQr2`7mH- #mi Bi ;2ib 2p2M #2ii2`5 uQmö`2 MQi HB /2}MBM; R M2r p bm#KK2`BX2UV

```
;/TnTQTn#v+QMiBM2Mibn#vv2` l@
    ;`QmT U+QMiBM2Mi- v2` V
    bmKK `Kx2Mn;/TS2`+ T U;/TS2`+ TV-
              b/n;/TS2`+ T U;/TS2`+ TV-
              K2 MnTQT U TQTV-
              b/nTQT UTQTVV
F #H2 U;/TnTQTn#v+QMiBM2Mibn#vv2 `VV
```

+QMiBM2Mi

v2 `

K2 Mn;/TS2`+ T

b/n;/TS2`+ T

K2 MnTQT

b/nTQT
7`B+
RN8k
Rk8j
N3j
98dyyRy
ejRd98y
7`B+
RN8d
Rj38
RRj8
8yNjyjj
dydey9k
7`B+
RNek
R8N3
R9ek
8dykk9d
dN8d898
7`B+
RNed
ky8y
k393
e99d3d8
3N388y8
7`B+
RNdk
kj9y
jk3d
djy8jde
RyRjy3jj

7`B+

RNdd

k83e

9R9k

3jk3yNd

RR838R39

RyXkXe //L2ro `B #H2bmiBi2? Kb

q? i B7 r2 r Mi2/ iQ //i?2b2 p Hm2b iQ Qm` Q`B;BM H/i 7` K2BMbi2/ +`2 iBM; M2r Q#D2+i\

6Q` i?Bb- r2 + MKmbi2i?2mM +iBQM- r?B+? Bm bKK`BxH2UVQ 2t+2Ti i? i Bi +`2 i2b M2r p `B #H2b BM i?2b K2/i 7` K2 i? i vQm T bb B BiX

```
; TKBM/2`nrBi?n2ti` n;p T`bU =@
    ;`QmTU#v+QMiBM2Mi-v=2 `V
    KmiU K2 Mn;/TS2`+ K2 M;/TS2`+ TV-
            b/n;/TS2`+ b/U;/TS2`+ TV-
            K2 MnTQK2 MUTQTV-
            b/nTQb/UTQTVV
F #H2 U; TKBM/2`nrBi?n2ti` np `bVV
```

+QmMi`v

v2 `

TQT

+QMiBM2Mi

HB721tT

;/TS2`+ T

K2 Mn;/TS2`+ T

b/n;/TS2`+ T

K2 MnTQT

b/nTQT

7;? MBbi M

RN8k

39k8jjj

Rjy *> Sh1_ RyX .h h_ La6P_J hAPL

bB
k3X3
ddN
8RN8
R3ej8
9kk3j88e
RXRj2Yy3
7;? MBbi M
RN8d
Nk9yNj9
bB
jyXj
3kR
8d33
RN8yd
9dj8eN33
RXk32Yy3
7;? MBbi M
RNek
Rykedy3j
bB
jkXy
38j
8dkN
Re9Re
8R9y9dej
RXje2Yy3
7;? MBbi M
RNed
RR8jdNee
bB

j9Xy

3je

8NdR

R9yej

8dd9djeR

RX8j2Yy3

7;? MBbi M

RNdk

RjydN9ey

bB

jeXR

d9y

3R3d

RNy33

e8R3yNdd

RXd92Yy3

7;? MBbi M

RNdd

R933yjdk

bB

j3X9

d3e

ddNR

RR3Re

dkk8dN3d

RXNk2Yy3

q2 + M mb2 HkmQim2UQV +`2 i2 M2r p `B #H2b T`BQ` iQ UQ` 2p2M 7i2`V bmKK `BxBM; i?2 BM7Q`K iBQMX

```
;/TnTQTn#v+QMiBM2Mib, T#v=2W=W` I@
    Kmi U2/Tn#BHHBQ/TS2`+TQfRyN V W=W
    ;`QmTU#v+QMiBM2Mi-W=2W` V
    bmKK `U Kx2Mn;/TS2`+K2TUM;/TS2`+ TV-
```

```
b/n;/TS2`+ T/U4;/TS2`+ TV-
K2 MnTQT/K2 4UTQTV-
b/nTQT/b/4UTQTV-
K2 Mn;/Tn#BHHBQM/K2 4U;/Tn#BHHBQMV-
b/n;/Tn#BHHBQM/b/4U;/Tn#BHHBQMVV
F #H22 /U2 U;/TnTQTn#v+QMiBM2Mibn#vv2 `VV
```

+QMiBM2Mi

v2 `

K2 Mn;/TS2`+ T

b/n;/TS2`+ T

K2 MnTQT

b/nTQT

K2 Mn;/Tn#BHHBQM

b/n;/Tn#BHHBQM

7`B+

RN8k

Rk8j

N3j

98dyyRy

ejRd98y

8XNN

RRX9

7`B+

RN8d

Rj38

RRj8

8yNjyjj

dydey9k

dXje

R9X8

7`B+

RNek

R 8 N 3

R 9 e k

8 d y k k 9 d

d N 8 d 8 9 8

3 X d N

R d X k

7 ` B +

R N e d

k y 8 y

k 3 9 3

e 9 9 d 3 d 8

3 N 3 8 8 y 8

R R X 9 9

k j X k

7 ` B +

R N d k

k j 9 y

j k 3 d

d j y 8 j d e

R y R j y 3 j j

R 8 X y d

j y X 9

7 ` B +

R N d d

k 8 3 e

9 R 9 k

3 j k 3 y N d

R R 8 3 8 R 3 9

R 3 X d y

j 3 X R

Kmi i2bX bmKK `Bx2

Ai + M #2 +QM7mbBM; iQ /2+B/2 r?2i?2` Kmi i2bX bmKK `Bx2 F2v /BbiBM+iBQM Bb r?2i?2` vQm r Mi i?2 QmiTmi iQ ? p2 QM2 `Qr 7Q` 2 +? `Qr BM i?2 Q`B;BM H / i 7` K2,

Ç Kmi i2 + `2 i2b M2r +QHmKMb rBi? b K Mv `Qrb b i?2 Q`B;BM H / i bmKK `Bx2 +`2 i2b / i 7` K2 rBi? b K Mv `Qrb b ;`QmTb

LQi2 i? i B7 vQm mb2 M ;;`2; iBQM 7mM+iBQM i? i `2im`Mb b BM;H2 p Hm2 rBi?BM Kmi i2UV rBi?Qmi mbBM; ;`QmTn#vUV i?2 bmKK `v Qp2` HH i?2 `Qrb BM i?2 / i 7` K2X

M/ B7 vQm mb2 M ;;`2; iBQM 7mM+iBQM i? i `2im`Mb b BM;H2 p Hm2 rBi?BM bmKK `Bx2UV rBi?Qmi mbBM; ;`QmTn#vUV i?2 bmKK `v Qp2` HH i?2 `Qrb BM i?2 / i 7` K2 Bb bBM;H2 ;`QmTVX

RyXkXd `` M;2 _Qrb `r BM?2

b H bi bi2T- H2iöb b v r2 r Mi iQ bQ`i i?2 `Qrb BM Qm` / i 7` K2 + p Hm2b BM +2`i BM +QHmKM`X q2 7Q` i?Bb i?2 QT2` iQ` /Q i?BbX 6Q` BMbi M+2- H2iöb Q`; MBx2 Qm` `2bmHib #v +QmMi `v UBM +`2 bBM; Q`/2`V- M/ i?2M #v v2 ` UBM /2b+2M/BM; Q`/2`VX

```
; TKBM/2`nrBi?n2ti` np T`b =W
  ;`QmTn#vU+QmMi`v- v2 `V
  Kmi2UK2 Mn;/TS2`+ T 4 K2 MU;/TS2`+ TV-
         b/n;/TS2`+ T 4 b/U;/TS2`+ TV-
         K2 MnTQT 4 K2 MUTQTV-
         b/nTQT 4 b/UTQTVV =W
  ``` M;2UbUv2 `V- +QmMi`vV
F #H2 U; TKBM/2`nrBi?n2ti` np `bVV
```

+QmMi`v

v2 `

TQT

+QMiBM2Mi

HB72 tT

;/TS2`+ T

K2 Mn;/TS2`+ T

b/n;/TS2`+ T

K2 MnTQT

b/nTQT
```

H;2`B

kyyd

jjjjkRe

7`B+

dkXj

ekkj

jy3N

jeR3

Rd3d8dej

k9NRddke

M;QH

kyyd

Rk9ky9de

7`B+

9kXd

9dNd

jy3N

jeR3

Rd3d8dej

k9NRddke

"2MBM

kyyd

3yd3jR9

7`B+

8eXd

R99R

jy3N

jeR3

Rd3d8dej

k9NRddke

"Qibr M

kyyd

RejNRjR

7`B+

8yXd

Rk8dy

jy3N

jeR3

Rd3d8dej

k9NRddke

"m`FBM 6 bQ

kyyd

R9jkekyj

7`B+

8kXj

RkRd

jy3N

jeR3

Rd3d8dej

k9NRddke

"m`mM/B

kyyd

3jNy8y8

7`B+

9NXe

9jy

jy3N

jeR3

Rd3d8dej

k9NRddke

RyXj/THv`M/ óLQM@ai M/ `/ 1p Hm iBQMô

uQm K v `mM +`Qbb i?2 i2`K óMQM@bi M/ `/ 2p Hm iBQMôX h?2 mb2 Q7 / i p `B #H2b rBi?Qmi [mQi2b `QmM/ i?2K Bb M 2t KTH2 Q7 i?BbX

q?v Bb i?Bb bi` M;2\

```
; T W=1W2H2+ +QMiBM2Mi- W=1W BV V
```

*QKT `2 BiiQ,

```
; T( -+U^+QMiBM2Mi^ `V)
; T( - +QMiBM2Mi)
```

"2+ m b2 +QMiBM2Mi ``2 MQi p `B #H2b Qm` +m``2Mi 2MpB`QMK2Mi- /Q2b bQK2 7 M+v bimz #2?BM/ i?2 b+2M2b iQ b p2 mb 7`QK ivTBM; i?2 [mQ

h?Bb Bb }M2 B7 vQm ? p2 / i M HvbBb rQ`F~Qr- #mi B7 vQm r Mi iQ r`B 7mM+iBQM i? i- 7Q` 2t KTH2- b2H2+ib M `#Bi` `v b2i Q7 +QHmKMb- vQm i`Qm#H2X

```
OO ?2`2^b ?2HT2` 7mM+iBQM i? i +QKTmi2b i?2 K2 M Q7 p `B #H2-
;`QmT2/nK2 7mM+iBQM- ;`QmTnp `- bmKK `vnp `V &
  / i W=W
    ;`QmTU#v`;`QmTnp W=VW
    bmKK `UK2 M K2 Ubm KK `vnp `VV
'
; T W=;`QmT2/nK2 +QMiBM2Mi- HB721tTV
; T W=;`QmT2/nK2 ^+QMiBM2Mi^- HB721tT^V
```

a22 i?2 M Q`b2THT `+F ;2b 7Q` ?Qr QM2 + M /2 H rBi? i?Bb T`Q#H2K BM i?Bb +QMi2ti Q7 mbBM; 7mM+iBQMbX

RyX9 *? HH2M;2b

*? HH2M;2 RX

Ib2/THvi`Q +`2 i2 / i 7` K2 +QMi BMBM; HB721tT +QHmKM /2 +M +QMiBM2MiX

*? HH2M;2 kX

Ib2/THvi`Q // +QHmKM iQ i?2 ; TKBM/2` / i b2i i? i +QMi BMb i?2 iQi H TQTmH iBQM Q7 i?2 +QMiBM2Mi Q7 2 +? Q#b2`p iBQM BM ;Bp2M v2 `X 6 i?2 }`bi Q#b2`p iBQM Bb 7;? MBbi M BM RN8k- i?2 M2r +QHmKM rQmH/ +Q TQTmH iBQM Q7 bB BM RN8kX

*? HH2M;2 jX

lb2/THvi`Q, U V // +QHmKM S+2H H2/ h/iB7i+QMi BMb i?2 /B7@
72`2M+2 #2ir22M i?2 Q#b2`p+iBQM Mo2bK2;/MS2`+QT7 i?2
+QMiBM2Mi BM i? i v2 `- U#V `` M;2 i?2 / i 7` K2 #v i?2 +QHmKM
i2/- BM /2b+2M/BM; Q`/2` UbQ i? i i?2 `2H iBp2Hv `B+?2bi +QmMi
}`biVX

?BM,iuQm KB;?i ?mpM2;i`QQ m#T2U7V Q`2 v`QmM;2XUV

+FMQrH2/;K2Mib

aQK2 Q7 i?2b2 K i2`B Hb BM i?Bb KQ/mH2 r2`2 / Ti2/ 7`QK,

Ç aQ7ir `2 * `TV2Mi`v
Ç _ #QQi+ KT i l* "2`F2H2v

*? Ti2` RR

hB/vBM;. i

1p2M #27Q`2 r2 +QM/m+i M HvbBb Q` + H+mH iBQMb- r2 M22/ iQ Tmi Qm
i?2 +Q``2+i 7Q`K iX h?2 ;Q H ?2`2 Bb iQ `2 `` M;2 K2bbv / i b2i BMiQ QM
BbB/X

h?2 irQ KQbi BKTQ`i Mi T`QT2`iB2b Q7 iB/v / i `2,

RV1 +? +QHmKM Bb p `B #H2X
kV1 +? `Qr Bb M Q#b2`p iBQMX

hB/v / i Bb 2 bB2` iQ rQ`F rBi? #2+ mb2 vQm ? p2 +QMbBbi2Mi r v Q7 `27
iQ p `B #H2b U b +QHmKM M K2bV M/ Q#b2`p iBQMb U b `Qr BM/B+2bVX
i?2M #2+QK2b 2 bB2` iQ K MBTmH i2- pBbm HBx2- M/ KQ/2HX

6Q` KQ`2 QM i?2 +QM+2Ti Q7 iB/v / i - `2 / >/H2v qB+F? KǶb T T2`

RRXR qB/2 pbX GQM; 6Q`K ib

óhB/v / i b2ib `2 HH HBF2 #mi 2p2`v K2bbv / i b2i Bb K2bbv BM Bib QrM r vXô >/H2v qB+F? K

h #mH `/ i b2ib+ M #2 `` M;2/ BM K Mv r vbX 6Q` BMbi M+2- +QMbB/2` i?2
#2HQrX 1 +?/ i b2i/BbTH vb BM7Q`K iBQM QM ?2 `i` i2b Q#b2`p2/ BM BM
+`Qbb j /Bz2`2Mi iBK2 T2`BQ/bX "mi i?2 / i `2 Q`; MBx2/ /Bz2`2MiHv BM
i #H2X

```
rB/2 I@ i X7` K2
  M K2 +U qBH#m]$2imM-B]`2;Q`V
  iBK2R +U ed 3y e9V-
  iBK2k +U 8e Ny 8yV-
  iBK2j +U dy ed RyRV
```

```
V
F #HU2rB/2V
```

M K2

iBK2R

iBK2k

iBK2j

qBH#m`

ed

8e

dy

S2imMB

3y

Ny

ed

:`2;Q`v

e9

8y

RyR

```
HQM; /@ X7` kU2
  M K2 +U]qBH#m]$2imMB]:]`2;Q`v]]qBH#m]$2imMB]:]`2;Q`v]]qBH#m]
  iBK2 +UR- R- R- k- k- k- j- j- jV-
  ?2 `i` i2 +U ed- 3y- e9- 8e- Ny- 8y- dy- ed- RyV
V
F #HU2HQM;V
```

M K2

iBK2

?2 `i` i2

qBH#m`

R

ed

S2imMB

R

3y

:`2;Q`v

R

e9

qBH#m`

k

8e

S2imMB

k

Ny

:`2;Q`v

k

8y

qBH#m`

j

dy

S2imMB

j

ed

:`2;Q`v

j

Ry

Zm2biBQMB +? QM2 Q7 i?2b2 /Q vQm 7QHHQr\Bb i?2

Mbr2` h?2 }`bi / i 7` K2 Ui?2 órB/2ô QM2V rQmH/ MQi #2 +QMbB/2`2/

#2+ mb2 p Hm2b UBX2X- ?2 `i ` i2V `2 bT`2 / +`Qbb KmHiBTH2 +QHmKM

q2 Q7i2M `272` iQ i?2b2 /Bz2`2Mi bi`m+im`2b b óHQM;ô pbX órB/2ô 7Q`

i?2 óHQM;ô 7Q`K i- vQm mbm HHv ? p2 R +QHmKM 7Q` i?2 Q#b2`p2/ p `B #

Qi?2` +QHmKMb `2 A. p `B #H2bX

6Q` i?2 órB/2ô 7Q`K i 2 +? `Qr Bb Q7i2M bBi2fbm#D2+ifT iB2Mi M/ vQm

KmHiBTH2 Q#b2`p iBQM p `B #H2b +QMi BMBM; i?2 b K2 ivT2 Q7 / i X h?2

2Bi?2` `2T2 i2/ Q#b2`p iBQMb Qp2` iBK2- Q` Q#b2`p iBQM Q7 KmHiBTH2

KBt Q7 #Qi?VX AM i?2 #Qp2 + b2- r2 ? / i?2 b K2 FBM/ Q7 / i U?2 `i ` i

2Mi2`2/ +`Qbb j /Bz2`2Mi +QHmKMb- +Q``2bTQM/BM; iQ i?`22 /Bz2`2Mi iB

FMB,i,`BM+Hm/2n;`UT?iB+4]BK;fiB/v`@7B;VRXTM;]

uQm K v }M/ / i BMTmi BM i?2 órB/2ô 7Q`K i iQ #2 bBKTH2`- M/ b
TTHB+ iBQMb K v T`272` órB/2ô 7Q`K i / i X >Qr2p2`- K Mv Q7 _
? p2 #22M /2bB;M2/ bbmKBM; vQm ? p2 óHQM;ô 7Q`K i / i X

RRXk hB/vBM; i?2 : TKBM/2` . i

G2iöb HQQF i i?2 bi`m+im`2 Q7 Qm` Q`B;BM H ; TKBM/2` / i 7` K2,

```
; T I@2 /X+Up i f; TKBM/2`@6Bp2u2 -. bii`X+M;p] b6 +iQh`_bl1V4
F #HU22 /U; TVV
```

+QmMi`v

v2 `

TQT

+QMiBM2Mi

HB721tT

;/TS2`+ T

7;? MBbi M

RN8k

39k8jjj

bB

k3X3

ddN

7;? MBbi M

RN8d

Nk9yNj9

bB

jyXj

3kR

7;? MBbi M

RNek

Rykedy3j

bB

jkXy

38j

7;? MBbi M

RNed

RR8jdNee

bB

j9Xy

3je

7;? MBbi M

RNdk

RjydN9ey

bB

jeXR

d9y

7;? MBbi M

RNdd

R933yjdk

bB

j3X9

d3e

Zm2biBQM; AM i?Bb / i r7B`/X2Q2HQM;

Mbr2`, h?Bb / i 7` K2 Bb bQK2r??2`2 BM #2ir22M i?2 Tm`2Hv õHQM
7Q`K ibX q2 ? p2 j óA. p+QB i#H2M2@QmU-iv`2 V M/ j óP#b2`@
p iBQM p `BT#H2B7 2U;ITS2`+V X

.2bTBi2 MQi ? pBM; GG Q#b2`p iBQMb BM R +QHmKM- i?Bb BMi
K F2b b2Mb2 ;Bp2M i? i HH j Q#b2`p iBQM p `B #H2b ? p2 /Bz2`2
? p2 b22M- K Mv Q7 i?2 7mM+iBQMb BM _ `2 Q7i2M p2+iQ` # b2/-
/Q MQi r Mi iQ /Q K i?2K iB+ H QT2` iBQMb QM p Hm2b rBi? /Bz2`2

PM i?2 Qi?2` ? M/- i?2`2 `2 bQK2 BMbi M+2b BM r?B+? Tm`2Hv
7Q`K i Bb B/2 H U2X;X THQiiBM;VX GBF2rBb2- bQK2iBK2b vQmöH
i? i Bb TQQ`Hv Q`; MBx2/- M/ vQm r Mi iQ +H2 M 2/ iQ

RRXj B/v `6mM+iBQMb

h? MF7mHHv- i?2 F ;2 rBHH ?2HT vQm 2{+B2MiHv i` Mb7Q`K vQ
`2; `/H2bb Q7 Q`B;BM H 7Q`K iX

```
O AMbi HH i?2 ]iB/v`] T +F ;2 UQMHv M2+2bb `v QM2 iBK2V
O BMbi HHXT +F ;2bU]iB/v`]V O LQi _mM

O GQ / i?2 ]iB/v`] T +F ;2 UM2+2bb `v 2p2`v M2r _ b2bbBQMV
HB#` `v UiB/v`V
```

RRXjXR i?2`

lMiBH MQr- r2öp2 #22M mbBM; i?2 MB+2Hv 7Q`K ii2/ Q`B;BM H ; TKBM/2
h?Bb / i b2i Bb MQi [mBi2 rB/2 M/ MQi [mBi2 HQM; Biöb bQK2i?BM; BM i?
#mi õ`2 Hö / i UBX2X Qm` QrM `2b2 `+? / i V rBHH M2p2` #2 bQ r2HH Q`; M
H2iöb bi `i rBi? i?2 rB/2 7Q`K i p2`bBQM Q7 i?2 ; TKBM/2` / i b2iX

```
; TnrB/2 l@ /X+Ujp i f; TKBM/2`nrB/2Xt`bpc]- b6 +iQ`bL a1
F #H2 U; TnrB/2VV
```

+QMiBM2Mi

+QmMi`v

;/TS2`+ TnRN8k

;/TS2`+ TnRN8d

;/TS2`+ TnRNek

;/TS2`+ TnRNed

;/TS2`+ TnRNdk

;/TS2`+ TnRNdd

;/TS2`+ TnRN3k

;/TS2`+ TnRN3d

;/TS2`+ TnRNNk

;/TS2`+ TnRNNd

;/TS2`+ Tnkyyk

;/TS2`+ Tnkyyd

HB721tTnRN8k

HB721tTnRN8d

HB721tTnRNek

HB721tTnRNed

HB721tTnRNdk

HB721tTnRNdd
HB721tTnRN3k
HB721tTnRN3d
HB721tTnRNNk
HB721tTnRNNd
HB721tTnkyyk
HB721tTnkyyd
TQTnRN8k
TQTnRN8d
TQTnRNek
TQTnRNed
TQTnRNdk
TQTnRNdd
TQTnRN3k
TQTnRN3d
TQTnRNNk
TQTnRNNd
TQTnkyyk
TQTnkyyd
7`B+
H;2`B
k99N
jyR9
k88R
jk9d
9R3j
9NRy
8d98
8e3R
8ykj
9dNd

8 k 3 3
e k k j
9 j X R
9 8 X d
9 3 X j
8 R X 9
8 9 X 8
8 3 X y
e R X 9
e 8 X 3
e d X d
e N X k
d R X y
d k X j
N k d N 8 k 8
R y k d y 3 8 e
R R y y y N 9 3
R k d e y 9 N N
R 9 d e y d 3 d
R d R 8 k 3 y 9
k y y j j d 8 j
k j k 8 9 N 8 e
k e k N 3 j d j
k N y d k y R 8
j R k 3 d R 9 k
j j j j j k R e
7 ` B +
M ; Q H
j 8 k R
j 3 k 3
9 k e N

88kj
89dj
jyyN
kd8d
k9jy
kek3
kkdd
kddj
9dNd
jyXy
jkXy
j9Xy
jeXy
jdXN
jNX8
jNXN
jNXN
9yXe
9RXy
9RXy
9kXd
9kjkyN8
98eRjeR
93keyR8
8k9d9eN
83N9383
eReked8
dyRej39
d3d9kjy
3dj8N33
N3d8yk9

Ry3eeRye
Rk9ky9de
7`B+
"2MBM
Ryej
Ney
N9N
Ryje
Ry3e
RykN
Rkd3
Rkke
RRNR
Rkjj
Rjdj
R99R
j3Xk
9yX9
9kXe
99XN
9dXy
9NXk
8yXN
8kXj
8jXN
89X3
89X9
8eXd
Rdj3jR8
RNk8Rdj
kR8R3N8

k9kdjj9
kdeR9yd
jRe3ked
je9Reyj
9k9jd33
9N3RedR
eyeey3y
dykeRRj
3yd3jR9
7`B+
"Qibr M
38R
NR3
N39
RkR8
kke9
jkR8
988R
ekye
dN89
3e9d
RRyy9
Rk8dy
9dXe
9NXe
8RX8
8jXj
8eXy
8NXj
eRX8
ejXe

ekXd

8kXe

9eXe

8yXd

99kjy3

9d9ejN

8Rkde9

88j89R

eRNj8R

d3R9dk

Ndyj9d

RR8RR39

Rj9keR9

R8je8je

Rejyj9d

RejNRjR

7`B+

"m`FBM 6 bQ

89j

eRd

dkj

dN8

388

d9j

3yd

NRk

Njk

N9e

Ryj3

RkRd

jkXy

R8k *> Sh1_ RRX hA.uAL: . h

j9XN
jdX3
9yXd
9jXe
9eXR
93XR
9NXe
8yXj
8yXj
8yXe
8kXj
99eNNdN
9dRj9Re
9NRNejk
8RkdNj8
89jj33e
833N8d9
eej98Ne
d83e88R
33d3jyj
Ryj8k39j
Rkk8RkyN
R9jkekyj
7`B+
"m`mM/B
jjN
j3y
j88
9Rj
9e9
88e

8ey

ekk

ejk

9ej

99e

9jy

jNXy

9yX8

9kXy

9jX8

99XR

98XN

9dX8

93Xk

99Xd

98Xj

9dX9

9NXe

k998eR3

keed8R3

kNeRNR8

jjjyN3N

j8kNN3j

j3j99R8

983y9Ry

8Rkeykj

83yNkje

eRkReRy

dykRyd3

3jNy8y8

h?2 }`bi bi2T iQr `/b ;2iiBM; Qm` MB+2 BMi2`K2/B i2 / i 7Q`K i Bb
p2`i 7`QK i?2 rB/2 iQ i?2 HQM; 7Q`K iX

h?2 7mM+iBQM rBHH õ; i?2`ö i?2 Q#b2`p iBQM p `B #H2b BMiQ b
#H2X h?Bb Bb bQK2iBK2b + HH2/ óK2HiBM;ô vQm` / i - #2+ mb2
7`QK rB/2 iQ HQM;X h?Qb2 / i rBHH #2 K2Hi2/ BMiQ irQ p `B #H2
p `B #H2 M K2b- M/ i?2 Qi?2` 7Q` i?2 p `B #H2 p Hm2bX

; TnHQM;; IT@ rB/2 W=W
; i?2 UQ#bivT2nv2 `- Q#bnp Hm2b- j,j3V
F #H2 U ; TnHQM;VV

+QMiBM2Mi

+QmMi`v

Q#bivT2nv2 `

Q#bnp Hm2b

7`B+

H;2`B

;/TS2`+ TnRN8k

k99N

7`B+

M;QH

;/TS2`+ TnRN8k

j8kR

7`B+

"2MBM

;/TS2`+ TnRN8k

Ryej

7`B+

"Qibr M

;/TS2`+ TnRN8k

38R

7`B+

"m`FBM 6 bQ

;/TS2`+ TnRN8k

89j

7`B+

"m`mM/B

;/TS2`+ TnRN8k

jjN

LQiB+2 i? ir2 Tmij `;mK2Mi2B UMQM i?2 iBQM,

RXh?2 M K2 i?2 M2r +QHmKM 7Q` i?2 QM2 bvA.2pv`B `#H2 U
kXh?2 M K2 7Q` i?2 M2r K H; K i2/ Q#b2`p iBQM Hm2`B #H2 U
jXh?2 BM/B+2b Q7 i?2 QH/ Q#b2`p iBQM pH`BB#H2 +QHmKMb j
i?`Qm;? j3V i? i r2 r Mi iQ ; i?2` BMiQ QM2 p `B #H2X LQiB+2 i? i r2
/QMöi r Mi iQ K2Hi /QrM +QHmKMb R M/ k- b i?2b2 `2 +QMbB/2`2/ ó
p `B #H2bX

q2 + M HbQ b2H2+i Q#b2`p iBQM p `B #H2b mbBM;,

- p `B #H2 BM/B+2b
- p `B #H2 M K2b UrBi?Qmi [mQi2bV
- t,xiQ b2H2+i HH p `B #H2b #2ir22M t M/ x
- @ iQ 2t+Hm/2
- bi `ibnrBi?Ut- B;MQ`2X+ b2H4 h_lK2V b i? i bi `ib rBi?
- 2M/bnrBi?Ut- B;MQ`2X+ b2H4 h_lK2V b i? i 2M/b rBi?
- +QMi BMbUt- B;MQ`2X+ b2H4 h_lK2V b i? i +QMi BM

a22 i?2 H2+i7UnV M+iB/Q H vB Q` KQ`2 QTiBQMbX

6Q` BMbi M+2- ?2`2 r2 /Q i?2 b K2 i?BM; rBi? bi `ibnrBi?UV M-
M/ UkV @Q2T2` iQ`,

```
O RX rBi? i?2 bi `ibnrBi?UV 7mM+iBQM
; TnHQM;; IT@nrB/2W=W
  ;i?2UQ#bivT2nv2 `- Q#bnp Hm2bU?TQV-
      bi `ibnrBU^HB721V-bi `ibnrBU^;/TS2`+V^
F #H2 U; TnHQM;VV
```

+QMiBM2Mi

+QmMi`v

Q#bivT2nv2 `

Q#bnp Hm2b

7`B+

H;2`B

TQTnRN8k

NkdN8k8

7`B+

M;QH

TQTnRN8k

9kjkyN8

7`B+

"2MBM

TQTnRN8k

Rdj3jR8

7`B+

"Qibr M

TQTnRN8k

99kjy3

7`B+

"m`FBM 6 bQ

TQTnRN8k

99eNNdN

7`B+

"m`mM/B

TQTnRN8k

k998eR3

```
O kX rBi? i?2 @ QT2` iQ`
; TnHQM;; IT@n rB/2 W=W
  ; i?2 UQ#bivT2nv2 `- Q#bnp Hm2b @ QmMi`vV
F #H?2 U; TnHQM;VV
```

+QMiBM2Mi

+QmMi`v

Q#bivT2nv2 `

Q#bnp Hm2b

7`B+

H;2`B

;/TS2`+ TnRN8k

k99N

7`B+

M;QH

;/TS2`+ TnRN8k

j8kR

7`B+

"2MBM

;/TS2`+ TnRN8k

Ryej

7`B+

"Qibr M

;/TS2`+ TnRN8k

38R

7`B+

"m`FBM 6 bQ

;/TS2`+ TnRN8k

89j

7`B+

"m`mM/B

;/TS2`+ TnRN8k

jjN

>Qr2p2` vQm +?QQb2 iQ /Q Bi- MQiB+2 i? i i?2 QmiTmi +QHH Tb2b HH Q bm`2/ p `B #H2b BMiQ irQ +QHmKMb, QM2 +QMi BMBM; i?2 M2r A.p `B #H +QMi BMBM; i?2 Q#b2`p iBQM p Hm2 7Q` i? i `QrX

RRXjXbk2T ` i2

uQmöHH MQiB+2 i? i BM Q#b i Q2M;v/2iimb2H v +QMi BMb k TB2+2b Q7 BM7Q`K iBQM- i?2 Q#b2`p iBQM Tb2 BM; i?2 TS2U+V TM/ i?2 v2 X

q2 + M mb2 b2T ` i2UV M+iBQM iQ bTHBi i?2 +? ` +i2` bi`BM;b BMiQ KmHiBTH2 p `B #H2b,

```
; TnHQM;nb2TTnHQM; W=W
  b2T ` U2Q#bivT2nvB2Mi Q +U^Q#bniv-T22 `V-b2T 4n] V W=W
  Kmi U22 ` 4bXBMi2U22 `VV
F #HU22 U; TnHQM;nb2TVV
```

+QMiBM2Mi

+QmMi`v

Q#bnivT2

v2 `

Q#bnp Hm2b

7`B+

H;2`B

;/TS2`+ T

RN8k

k99N

7`B+

M;QH

;/TS2`+ T

RN8k

j8kR

7`B+

"2MBM

;/TS2`+ T

RN8k

Ryej

7`B+

"Qibr M

;/TS2`+ T

RN8k

38R

7`B+

"m`FBM 6 bQ

;/TS2`+ T

RN8k

89j

7`B+

"m`mM/B

;/TS2`+ T

RN8k

jjN

RRXjXbjT`2 /

h?2 QTTQb, Bi?2 QUB bT`2 /XV Ai bT`2 /b Qm` Q#b2`p iBQM p `B #H2b # +F QmiiQ K F2 rB/2` i #H2X q2 + M mb2 i?Bb ;7 nmM+iBQMV iQ bT`2 / Qm` iQ i?2 Q`B;BM H óK2/BmKô 7Q`K iX

```
; TnK2/BmK; T@HQM;m b=2 W=W
  bT`2 U/Q#bnivT2- Q#bnp Hm2bV
F #H22 U; TnK2/BmKVV
```

+QMiBM2Mi

+QmMi`v

v2 `

;/TS2`+ T

HB721tT

TQT

7`B+

H;2`B

RN8k

k99N

9jXR

NkdN8k8

7`B+

H;2`B

RN8d

Rey *> Sh1_ RRX hA.uAL: . h

jyR9

98Xd

Rykdy38e

7`B+

H;2`B

RNek

k88R

93Xj

RRyyyN93

7`B+

H;2`B

RNed

jk9d

8RX9

RkdeyNN

7`B+

H;2`B

RNdk

9R3j

89X8

R9deyd3d

7`B+

H;2`B

RNdd

9NRy

83Xy

RdR8k3y9

HH r2 M22/ Bb bQK2 [mB+F }t2b iQ K F2 i?Bb / i b2i B/2MiB+ H iQ
; TKBM/2i` b2i,

```
; T I@2 /X+Ujp i f; TKBM/2`@6Bp2u2 V. i X+bp]
F #HU?22 U; TVV
```

+QmMi`v

v2 `

TQT

+QMiBM2Mi

HB721tT

;/TS2`+ T

7;? MBbi M

RN8k

39k8jjj

bB

k3X3

ddN

7;? MBbi M

RN8d

Nk9yNj9

bB

jyXj

3kR

7;? MBbi M

RNek

Rykedy3j

bB

jkXy

38j

7;? MBbi M

RNed

RR8jdNee

bB

j9Xy

3je

7;? MBbi M

RNdk

RjydN9ey

bB

jeXR

d9y

7;? MBbi M

RNdd

R933yjdk

bB

j3X9

d3e

```
O `2 `` M;2 +QHmKMb
; TnK2/BmK; T@K2/BmMKK(+U); TV)
F #HU22 /U; TnK2/BmKVV
```

+QmMi`v

v2 `

TQT

+QMiBM2Mi

HB721tT

;/TS2`+ T

H;2`B

RN8k

NkdN8k8

7`B+

9jXR

k99N

H;2`B

RN8d

Rykdy38e

7`B+

98Xd

jyR9

H;2`B

RNek

RRyyyN93

7`B+

93Xj

k88R

H;2`B

RNed

Rkdey9NN

7`B+

8RX9

jk9d

H;2`B

RNdk

R9deyd3d

7`B+

89X8

9R3j

H;2`B

RNdd

RdR8k3y9

7`B+

83Xy

9NRy

```
O  `` M;2 #v +QmMi`v- +QMiBM2Mi-  M/ v2 `
; TnK2/BmK; T@K2/BmK=W
  `` M;U2+QmMi`v-+QMiBM2Mi-v2 `V
F #HU22 U; TnK2/BmKVV
```

+QmMi`v

v2 `

TQT
+QMiBM2Mi
HB721tT
;/TS2`+ T
7;? MBbi M
RN8k
39k8jjj
bB
k3X3
ddN
7;? MBbi M
RN8d
Nk9yNj9
bB
jyXj
3kR
7;? MBbi M
RNek
Rykedy3j
bB
jkXy
38j
7;? MBbi M
RNed
RR8jdNee
bB
j9Xy
3je
7;? MBbi M
RNdk
RjydN9ey

bB

jeXR

d9y

7;? MBbi M

RNdd

R933yjdk

bB

j3X9

d3e

q? ir2 Dmbi iQH/ vQm rBHH #2+QK2 Q#bQH2i2

; i?2`M/bT`2 /`2 #2BM; `2THB+2/ #v HQM;2`pQinrB/2` iB/v` RX-yXyB+? mb2 B/2 b+7`iQTK+F2;2 iQ K F2 `2b? TBM; 2 bB2` iQ i?BMF #QmiX AM 7mim`2 +H bb2b- r2öHH KB;` i2 iQ i?Qb2 7mM+iBQMb

RRX9 JQB2vp2`b2

/THv`M/iB/v`? p2 K Mv KQ`2 7mM+iBQMb iQ ?2HT vQm r` M;H2 M/ K MBTmH vQm` / i X a22 i?2 M;HBM; *?2 i aQ?2Ki`2X

h?2`2 `2 bQK2 Qi?2` mb27mH 7mM+iBQMb BM i?2

- ;;THQ7kQ` THQiiBM; Ur2öHH +Qp2` i?Bb BM i?2 oBbm HBx iBQM KQ/mH
- `2 /` M/? p2TMQ` `2 /BM; BM / i
- Tm``7Q` rQ`FBM; Bi2` iBQMbX
- bi`BM-Hm#`B/-7Q`+ 7bQ` K MBTmH iBM; bi`BM;b- / i2b- M/ 7 +iQ`b- `2bT2+iBp2Hv
- K Mv K Mv KQ`25 h F2 TiB/Fvp2`b2 ;Bi?m# T ;2

S`Q hBTh,Q BMbi HH M/ HQ / i?2 +Q`2 iB/vp2`b2 vF +F ;2b UBM+Hm/2b /THv` M/;THQ-ikKQM; Qi?2`bV- i`v,

```
OO LPh `mM
O BMbi HHXT +F ;2bU]iB/vp2`b2]V
HB#` Uvi B/vp2`b2V
```

RRX8 *? HH2M;2b

*? HH2M;2 RX

am#b2i i?2 `2bmHib 7`QK *? HH2M;2 Oj UQ7 i?2 T`2pBQmb +? Ti2
i?2 +QmMiv2 - M//TS2`+ Tn/+BQ7HmKMbiB/VbTmi Bi BM rB/2
7Q`K i bQ i? i +QmMi`B2b `2 `Qrb M/ v2 `b `2 +QHmKMbX

*? HH2M;2 kX

LQr im`M i?2 / i 7` K2 #Qp2 # +F BMiQ i?2 HQM; 7Q`K i rBi? i?`2
+QmMiv2 - M//TS2`+ TnXB77

+FMQrH2/;K2Mib

aQK2 Q7 i?2b2 K i2`B Hb BM i?Bb KQ/mH2 r2`2 / Ti2/ 7`QK,

- aQ7ir `2 * `T2Mi`v
- _ #QQi+ KT i l* "2`F2H2v

HB#` Uv F #H21ti` V

*? Ti2` Rk

_2H iBQM H . i

Aiöb` `2i? i/i M HvbBb BMpQHp2b QMHv bBM;H2i #H2 Q7/i X hvTB
? p2 K Mv i #H2b Q7/i - M/ vQm Kmbi +QK#BM2 i?2K iQ Mbr2` i?2 [m2biB
i? i vQmö`2 BMi2`2bi2/ BMX *QHH2+iBp2Hv- Km2HHB@ H2 i #H2b Q7/i `2
iBQM H /# 2+ mb2 Bi Bb i?2 `2H iBQMb- MQi Dmbi i?2 BM/BpB/m H /i b2ib
`2 BKTQ`i MiX

LQi2 i? i r?2M r2 b v `2H iBQM H /i # b2 ?2`2- r2 `2 `272``BM; iQ ?Qr i?
/i `2 bi`m+im`2/- MQi iQ i?2 mb2 Q7 Mv 7 M+v bQ7ir `2X

RkXR q?v _2H iBQM H . i

bbQ+B H b+B2MiBbib- r2ö`2 Q7i2M rQ`FBM; rBi? /i +`Qbb /Bz2`2Mi H2p
bBbX h?2 K BM T`BM+BTH2 Q7 `2H iBQM H /i Bb i? i2 +? i #H2 Bb bi`m+i
i?2 b K2 Q#b2`p iBQM H mMBiX

q?v Bb i?Bb BKTQ`i Mi\ *?2+F Qmi i?2 7QHHQrBM; /i X

```
K2bbv /@ X7` K2
 +QmMiv U4 jeyje jeyj3 jeyjN jey9- L - jdyyR jdyyk jdyyj V-
 bi i2 4 U ^Lu ^^Lu ^^Lu ^L - L - ^o ^ ^o ^ ^o V-
 +MivnTQT U4 j3Rdd j89kkN jNk9NkyR 9j9jk L - jkk3kN 9 9N9 Nj3j3 3V-
 bi i2nTQT U4 jjkyN yj9jjkyN yjL - 9jjkyN yj9jjkyN yjdRdjy ydRdjy ydRdjy yV-
 `2;BQM U4R- R- R- R- R- j- j- 9V
V

F #H2 UK2bbvV
```

+QmMiv

bi i2

+MivnTQT
bi i2nTQT
`2;BQM
jeyjd
Lu
j3Rddj8
9jjkyNyj
R
jeyj3
Lu
9kkNNN
9jjkyNyj
R
jeyjN
Lu
jk9Nky
L
R
jey9y
L
R9j9jk
9jjkyNyj
R
L
L
L
9jjkyNyj
R
jdyyR
o
jkk3kNy

RkXRX q>u _1G hAPL G . h ReN

dRdjyyy

j

jdyyk

o

99N9NN

dRdjyyy

j

jdyyj

o

j3j333

dRdjyyy

9

q? i K2bb5 >Qr + M i?2 TQTmH iBQM Q7 i?2 bi i2 Q7 L2r uQ`F #2 9j KBHH
7Q` QM2 +QmMiv #mi óKBbbBM;ô 7Q` MQi?2`\ A7 i?Bb Bb / i b2i Q7 +Qm
/Q2b Bi K2 M r?2M i?2 ó+QmMivô }2H/ Bb KBbbBM;\ A7 `2;BQM Bb bQK2i
*2Mbmb `2;BQM- ?Qr + M irQ +QmMiB2b BM i?2 b K2 bi i2 #2 BM /Bz2`2Mi
M/ r?v Bb Bi i? i HH i?2 +QmMiB2b r?Qb2 +Q/2b bi `i rBi? je `2 BM L2r u
2t+2Ti 7Q` QM2- r?2`2 i?2 bi i2 Bb mMFMQrM\

A7 r2 7QHHQr i?2 T`BM+BTH2b Q7 `2H iBQM H / i - 2 +? ivT2 Q7 Q#b2`p
b?QmH/ 7Q`K i #H2X

Ç +QmMiB2b +QMi BMb / i QM +QmMiB2bX
Ç bi i2b +QMi BMb / i QM bi i2b

aQ Qm` / i b?QmH/ HQQF HBF2,

```
+QmMiB2b / i @7` K2
+QmMiv+U4jeyjd- jeyj3- jeyjN- jey9y- jdyyR- jdyyk- jdyyj V-
bi i2 4+U ^Lu^- ^Lu^- ^Lu^- ^Lu^- ^^o ^- ^^o ^- ^^o ^V-
+QmMivnTQT 4+U3Rdd- j9kkN- jN9Nky- R9j9- jkjk3k- N99N9N- j3j333V- bi`BM;b b6 +iQ6
V
F #H2U+QmMiB2bV
```

+QmMiv

bi i2

+QmMivnTQT

jeyjd

Lu

j3Rddj8

jeyj3

Lu

9kkNNN

jeyjN

Lu

jk9Nky

jey9y

Lu

R9j9jk

jdyyR

o

jkk3kNy

jdyyk

o

99N9NN

jdyyj

o

j3j333

```
bi i2b I@i X7` kU2
  bi i2 4+U]Lu-]]o ]V-
  bi i2nTQT+U4jjkyN-yjdRdjyV-y
  `2;BQM+U4R- jV-bi`BM;b b6 +iQ6`b 4
V
F #HU2bi i2bV
```

bi i2

bi i2nTQT

`2;BQM

Lu

9jjkyNyj

R

o

dRdjyyy

j

*QmMiv TQTmH iBQM Bb T`QT2`iv Q7 +QmMiv- bQ Bi HBp2b BM i?2 +Q
ai i2 TQTmH iBQM Bb T`QT2`iv Q7 bi i2- bQ Bi + MMQi HBp2 BM i?2 +Q
A7 r2 ? / T M2H / i QM +QmMiB2b- r2 rQmH/ M22/ b2T ` i2 i #H2b 7Q` i?BM
p `v i i?2 +QmMiv H2p2H UHBF2 bi i2V M/ i?BM;b i? i p `v i i?2 +QmMiv@v
UHBF2 TQTmH iBQMVX

LQr i?2 K#B;mBiv Bb ;QM2X 1p2`v +QmMiv ? b TQTmH iBQM M/ bi i2X
bi i2 ? b TQTmH iBQM M/ `2;BQMX h?2`2 `2 MQ KBbbBM; bi i2b- MQ
+QmMiB2b- M/ MQ +QM~B+iBM; /2}MBiBQMbX h?2 / i # b2 Bb b2H7@/Q+m

RkXk E2vb

h?2 p `B #H2b mb2/ iQ +QMM2+i 2 +? T B` Q7 i #H2b `2 + HH2/ F2vbX F2v Bb
p `B #H2 UQ` b2i Q7 p `B #H2bV i? i mMB[m2Hv B/2MiB}2b M Q#b2`p iBQM
mMB[m2 B/2MiB}2`X

Ç E2vb `2 +QKTH2i2X h?2v M2p2`i F2 QM KBbbBM; p Hm2bX
Ç E2vb `2 mMB[m2X h?2v `2 M22` /mTHB+ i2/ +`Qbb `Qrb Q7 i #H2X

AM bBKTH2 + b2b- bBM;H2 p `B #H2 Bb bm{+B2Mi iQ B/2MiB7v M Q#b2`p
2t KTH2 #Qp2- 2 +? +QmMiv Bb mMB[m2Hv B/2MiB}2/ #v Bib +QmMiv@MK2 +Q/2 B/2MiB}2`Vc
2 +? bi i2 Bb B/2MiB}2/ #v Bib bi i2 Q` rBi? @H2ii2` bi`BM;VX

h?2`2 `2 irQ ivT2b Q7 F2vb,

Ç T`BK `v F2v mMB[m2Hv B/2MiB}2b M Q#b2`p iBQM BM Bib QrM i #H2X
2t KTH2- +QmMiB2b0 +QmMiv F2v T`BK `v F2v #2+ mb2 Bi mMB[m2Hv B/2MiB}2b
2 +? +QmMiv BM i?2 +QmMiB2b i #H2X

Ç 7Q`2B;M F2v mMB[m2Hv B/2MiB}2b M Q#b2`p iBQM BM MQi?2` i #H2X
2t KTH2- i?2 +QmMiB2b0bi i2 Bb 7Q`2B;M F2v #2+ mb2 Bi TT2 `b
+QmMiB2b i #H2 r?2`2 Bi K i+?2b 2 +? +QmMiv iQ mMB[m2 bi i2X

T`BK `v F2v M/ i?2 +Q``2bTQM/BM; 7Q`2B;M F2v BM MQi?2` i #H2 7Q`K
iBQMX

aQK2iBK2b i #H2 /Q2bMöi ? p2 M 2tTHB+Bi T`BK `v F2v, 2 +? `Qr Bb M
p iBQM- #mi MQ +QK#BM iBQM Q7 p `B #H2b `2HB #Hv B/2MiB}2b BiX A7
T`BK `v F2v- Biöb mb27mH iQ //M2 rBi? `QrnMmK#2`UV Bb
+ HH2/ bm``Q; i2 F2vX

RkXj CQBMb

. i biQ`2/ BM i?2 7Q`K r2 ? p2 QmiHBM2/ MQ`KpH2xBx2/ +QMbB/2`2/ ;2M2` H- r2 b?QmH/ i`v iQ F22T / i MQ`K HBx2/ b 7 ` BMiQ i?2 +Q r2 + MX aiQ`BM; MQ`K HBx2/ / i K2 Mb vQm` / i rBHH #2 2 bB2` i M/ BirBHH #2 ? `/2` iQ K F2 +QbiHv KBbi F2bX

i bQK2 TQBMi- ?Qr2p2`- r2ö`2 ;QBM; iQ r Mi iQ #`BM; i?2 UQ` iQ;2i?2` iQ T`Q/m+2 bBM;H2 / i 7` K2- M/ +QM/m+i M HvbBb QM

G2iöb b v r2 r Mi2/ iQ K2`;2 i?2 QpHHQrb vQm iQ +QK#BM2 p `B #H2b 7`QK i?2 irQ i #H2bX Ai }`bi K i+?2b Q#b2`p iBQMb #v i +QTB2b +`Qbb p `B #H2b 7`QK QM2 i #H2 iQ i?2 Qi?2`X

h?2`2 `2 7Qm` DQBM QTiBQMb,

RX MBMM2` DQBM, F22Tb Q#b2`p iBQMb i? i TT2 ` BM #Qi? i #H2bX
kX H27i DQBM, F22Tb HH Q#b2`p iBQMb BM
jX `B;?i DQBM, F22Tb HH Q#b2`p iBQMb BM
9X 7mHH DQBM, F22Tb HH Q#b2`p iBQMb BM

h?2 KQbi +QKKQMHv mb2/ DQBM Bb i?2 H27i DQBM, r?B+? mb2 i?Bb r?2M2p2` vQm HQQF mT //BiBQM H / i 7`QK MQi?2` i #H2- #2+ mb2 Bi T`2b2`p2b Q#b2`p iBQMb 2p2M r?2M i?2`2 BbMöi K i+?X h?2 H27i DQBM KTH2- M/v TmHHb BM p `B #H2b 7`QK ` BM; HH i?2 Q#b2`p iBQMb QM

G2iöb b v r2 r Mi iQ +QK#BM2 i?2 irQ i2bi #H2b r2 +`2 i2/ 2 `HB2`X

```
+QmMiB2bnbi i2b >W HW
  H27inDQBM U#v 4 ]bi i2]V

F #H2U+QmMiB2bnbi i2bV
```

+QmMiv

bi i2

+QmMivnTQT

bi i2nTQT

`2;BQM

jeyjd

Lu

j3Rddj8

9jjkyNyj

R

RkXjX CPALa Rdj

jeyj3
Lu
9kkNNN
9jjkyNyj
R
jeyjN
Lu
jk9Nky
9jjkyNyj
R
jey9y
Lu
R9j9jk
9jjkyNyj
R
jdyyR
o
jkk3kNy
dRdjyyy
j
jdyyk
o
99N9NN
dRdjyyy
j
jdyyj
o
j3j333
dRdjyyy
j
LQiB+2 i?2`2 `2 irQ M2bri+iQ2HHTmQKTM b2„BQXM

h?2 H27i DQBM b?QmH/ #2 vQm` /27 mHi DQBM, mb2 Bi mMH2bb vQ
iQ T`272` QM2 Q7 i?2 Qi?2`bX

RkX9 .2}MBM; E2vb

AM i?2 2t KTH2 #Qp2- i?2 irQ i #H2b r2`2 DQBM2/ #v bBM;H2 p `
p `B #H2 ? b i?2 b K2 M K2 BM #Qi? i #H2bX h? i #vQMbi` BMir b2
4]F2vX]

uQm + M mb2 Qi?2` #vi#H2b ? mb2 #vQM7Q`+i i?2 i #H2b BM Qi?2` r vb,

RX h?2 /27 #vHi- LlG Gmb2b HH p `B #H2b i? i TT2 `BM #Qi? i #H2b
r2 KB;?i + HH óM im` H DQBMôX

6Q` 2t KTH2- H2iöb b v r2 r Mi2/ iQ //; T+QHHm/2`M bLQ i?2
i? i 2M+Q/2b i?2 `2;BK2 ivT2 Q7 2 +? +QmMi`v@v2 ` Q#b2`p iBQM
/ i 7`QK TQHBiv i Ab2iX

; TKBM/2`` 2@/X+U / i f; TKBM/2`X+bb`BM;b b6 +iQ`b 4 6V
TQHBiv `2@/X+U / i fTQHBivnbm-#X+bbpM];b b6 +iQ`b 4 6V
F #H22 / UTQHBivVV

+QmMi`v

v2 `

TQHBivk

7;? MBbi M

R3yy

@e

7;? MBbi M

R3yR

@e

7;? MBbi M

R3yk

@e

7;? MBbi M

R3yj

@e

7;? MBbi M

R3y9

@e

7;? MBbi M

R3y8

@e

q2ö`2 MQr `2 /viQ DQBMi?2i #H2bX h?2 +QKK+QmMi M2`b #2ir22Mi?2K `2 M/v2 ;`

; TR I@TKBM/2` W=W
H27inDQTBMUTQHBivV
O= CQBMBM;- #v 4 +U]+QmMi`v]- ]v2 `]V

F #H22 U; TRVV

+QmMi`v

v2 `

TQT

+QMiBM2Mi

HB721tT

;/TS2`+ T

TQHBivk

7;? MBbi M

RN8k

39k8jjj

bB

k3X3

ddN

@Ry

7;? MBbi M

RN8d

Nk9yNj9

bB

jyXj

3kR

@Ry
7;? MBbi M
RNek
Rykedy3j
bB
jkXy
38j
@Ry
7;? MBbi M
RNed
RR8jdNee
bB
j9Xy
3je
@d
7;? MBbi M
RNdk
RjydN9ey
bB
jeXR
d9y
@d
7;? MBbi M
RNdd
R933yjdk
bB
j3X9
d3e
@d

kX +? ` +i2` p2#vi Q4`-+U]t]- ]Xv]N?Bb Bb HBF2 M im` H DQBM- #m
mb2b QMHv bQK2 Q7 i?2 +QKKQM p `B #H2bX

jX M K2/+? ` +i2` p#2v+4Q+`U] ] 4 ]X#]hV?BbrBHH K i+?p `B #H2 BMi #H2Q p `B #H2 BMi #vX2h?2 p `B #H2 rbBi?HQK2 mb2/ BM i?2 QmiTmiX

6Q` 2t KTH2- H2iöb // MQi?2`; pTK`BMi#H2` bQ2iQmiT`?vbB+ H BMi2;`Biv `B;?ib 7`QK i?2 *A_A / i b2iX

```
+B`B I@ /X+U]pi f+B`Bnbm#-Xbi`BM;b b6 +iQ`6V 4
F #H22 U+B`BVV
```

*h_u

u1 _

S>uaALh

7;? MBbi M

RN3R

y

7;? MBbi M

RN3k

y

7;? MBbi M

RN3j

y

7;? MBbi M

RN39

y

7;? MBbi M

RN38

y

7;? MBbi M

RN3e

y

"Qi? / i b2ib ? p2 +QmMi`v M/ v2 ` +QHmKMb- #mi i?2v `2 M K2/ /Bz2`2M

```
; Tk I@ TRW=W
  H27inDUQ+BM B#v 4+U]+QmMi`v]]*h_u+]]v2 `]4 ]u1 _]V

F #H22 U; TkVV
```

12.5 Duplicate Keys

So far we have assumed that the keys are unique. But that's not always the case. For example,

```
x <- data.frame(key = c(1, 2),
                val_y = c("x1", "x2"))

y <- data.frame(key = c(1, 2, 2, 1),
                val_x = c("y1", "y2", "y3", "y4"))

left_join(x, y, by = "key")
#>   key val_y val_x
#> 1   1    x1    y1
#> 2   1    x1    y4
#> 3   2    x2    y2
#> 4   2    x2    y3
```

Notice that this can sometimes cause unintended duplicates.

Acknowledgements

This page is in part derived from the following sources:

1. R for Data Science licensed under Creative Commons Attribution-NonCommercial-NoDerivs 3.0
2. Gentzkow, Matthew and Jesse M. Shapiro. 2014. Code and Data for the Social Sciences: A Practitioner's Guide.

Chapter 13

Plotting

> "Make it informative, then make it pretty"

There are two major sets of tools for creating plots in R:

- 1. base, which come with all R installations.
- 2. ggplot2, a stand-alone package.

Note that other plotting facilities do exist (notably **lattice**), but base and `ggplot2` are by far the most popular.

13.1 The Dataset

For the following examples, we will be using the gapminder dataset we've used previously. Gapminder is a country-year dataset with information on life expectancy, among other things.

```
gap <- read.csv("data/gapminder-FiveYearData.csv", stringsAsFactors = F)
```

13.2 R Base Graphics

The **basic** call takes the following form:

```
plot(x=, y=)
```

```
plot(x = gap$gdpPercap, y = gap$lifeExp)
```

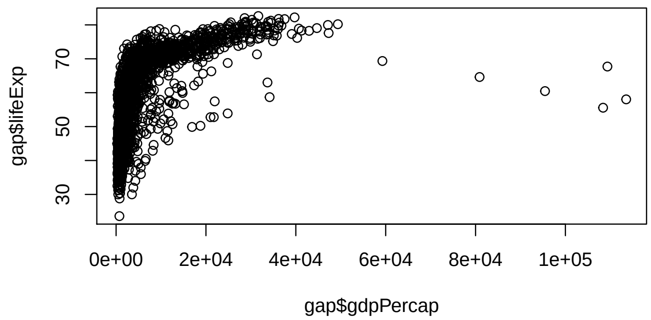
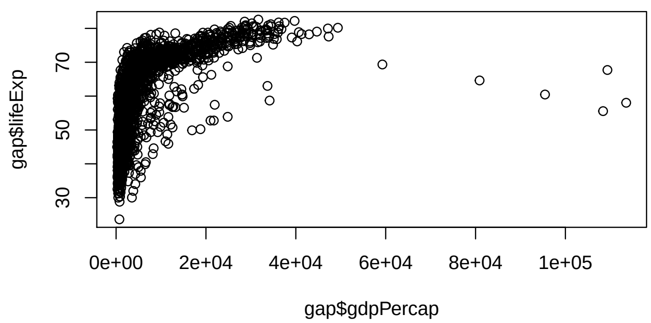


Figure 13.1:



13.2.1 Scatter and Line Plots

The “type” argument accepts the following character indicators.

- “p” – point/scatter plots (default plotting behavior)

```
plot(x = gap$gdpPercap, y = gap$lifeExp, type="p")
```

- “l” – line graphs

```
# Note that "line" does not create a smoothing line, just connected points
plot(x = gap$gdpPercap, y = gap$lifeExp, type="l")
```

- “b” – both line and point plots

```
plot(x = gap$gdpPercap, y = gap$lifeExp, type="b")
```

13.2.2 Histograms and Density Plots

Histograms display the frequency of different values of a variable.

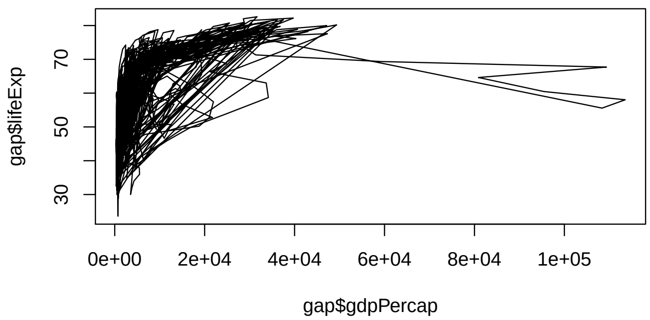


Figure 13.2:

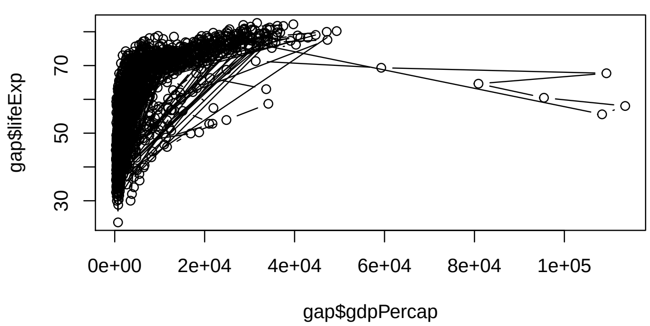
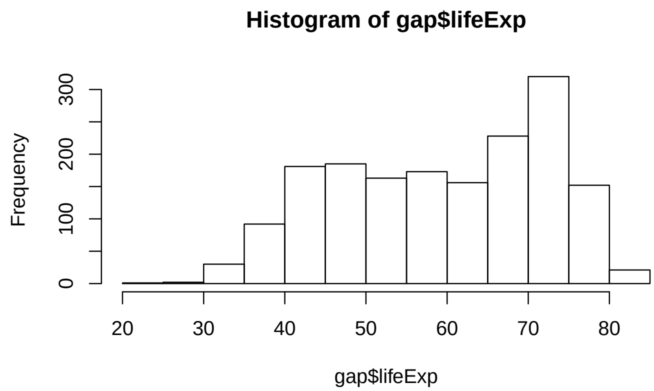


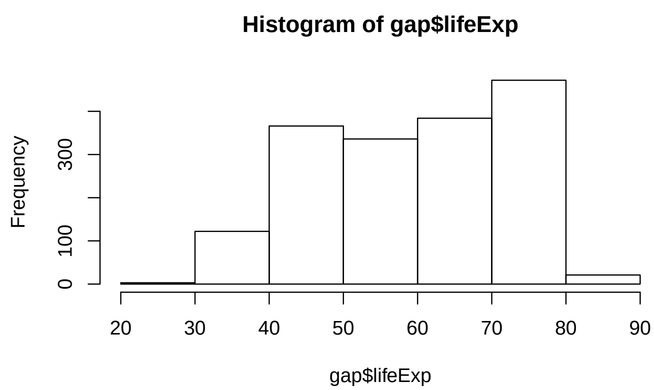
Figure 13.3:

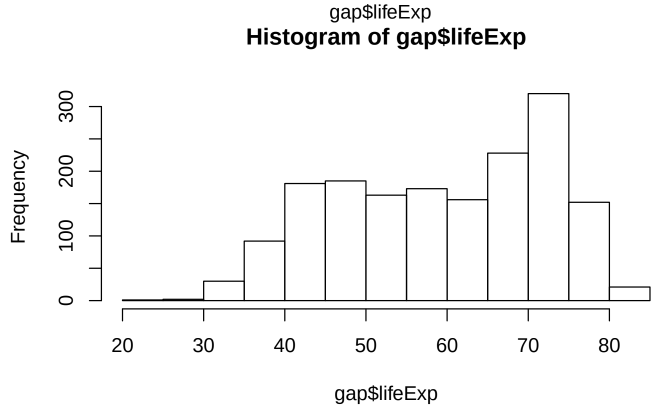
```
hist(x=gap$lifeExp)
```



Histograms require a `breaks` argument, which determine the number of bins in the plot. Let's play around with different `breaks` values.

```
hist(x=gap$lifeExp, breaks=5)
hist(x=gap$lifeExp, breaks=10)
```





Density plots are similar, they visualize the distribution of data over a continu-

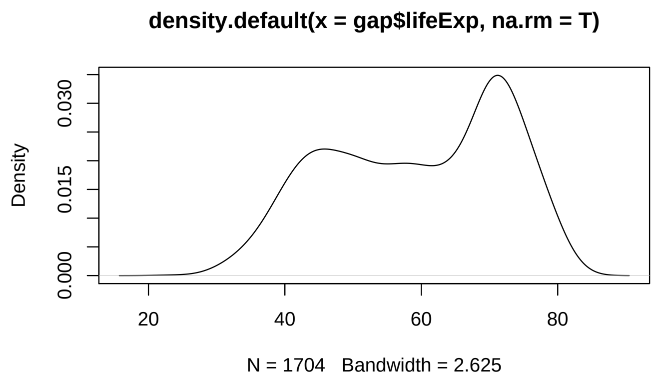


Figure 13.4:

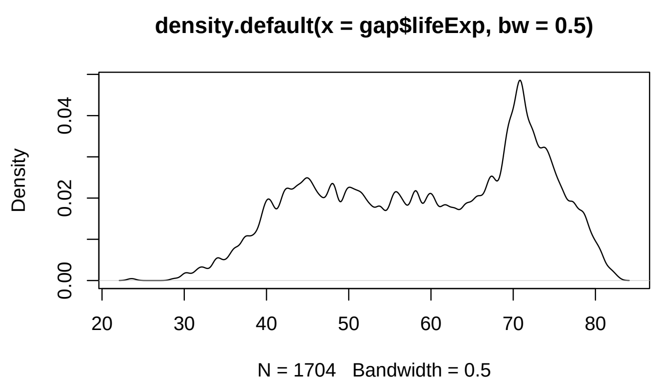
ous interval.

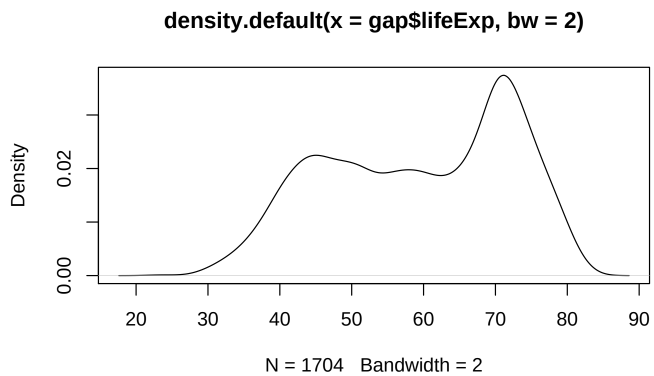
```
# Create a density object (NOTE: be sure to remove missing values)
age.density <- density(x=gap$lifeExp, na.rm=T)

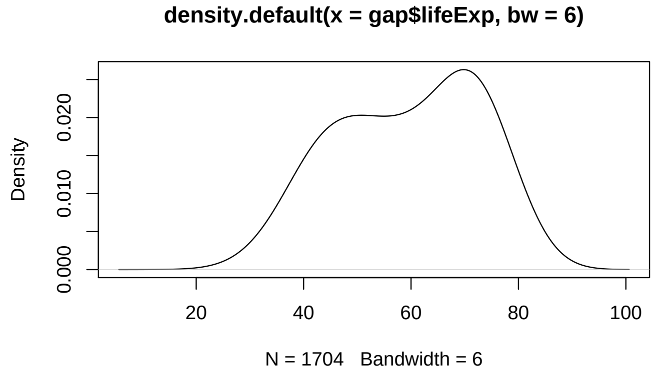
# Plot the density object
plot(x=age.density)
```

Density passes a `bw` parameter, which determins the plot's "bandwidth".

```
# Plot the density object, bandwidth of 0.5
plot(x=density(x=gap$lifeExp, bw=.5))
# Plot the density object, bandwidth of 2
plot(x=density(x=gap$lifeExp, bw=2))
# Plot the density object, bandwidth of 6
plot(x=density(x=gap$lifeExp, bw=6))
```







13.2.3 Labels

Here's the basic call with popular labeling arguments:

```
plot(x=, y=, type="", xlab="", ylab="", main="")
```

From the previous example...

```
plot(x = gap$gdpPercap, y = gap$lifeExp, type="p", xlab="GDP per cap", ylab="Life Expe
```

13.2.4 Axis and Size Scaling

Currently it's hard to see the relationship between the points due to some strong outliers in GDP per capita. We can change the scale of units on the x-axis using scaling arguments.

Here's the basic call with popular scaling arguments

```
plot(x=, y=, type="", xlim=, ylim=, cex=)
```

From the previous example...

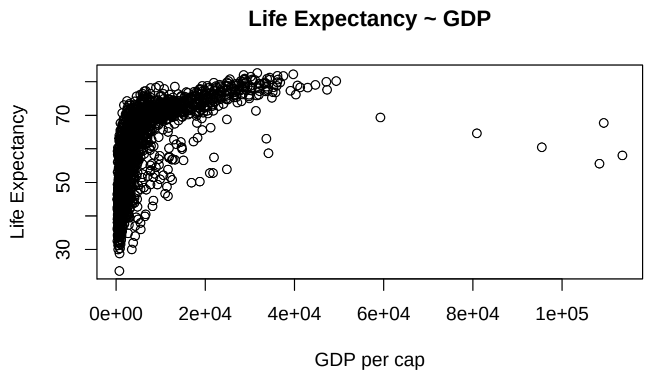


Figure 13.5:

```
# Create a basic plot
plot(x = gap$gdpPercap, y = gap$lifeExp, type="p")
# Limit gdp (x-axis) to between 1,000 and 20,000
plot(x = gap$gdpPercap, y = gap$lifeExp, xlim = c(1000,20000))
# Limit gdp (x-axis) to between 1,000 and 20,000, increase point size to 2
plot(x = gap$gdpPercap, y = gap$lifeExp, xlim = c(1000,20000), cex=2)
# Limit gdp (x-axis) to between 1,000 and 20,000, decrease point size to 0.5
plot(x = gap$gdpPercap, y = gap$lifeExp, xlim = c(1000,20000), cex=0.5)
```

13.2.5 Graphical Parameters

We can change the points with a number of graphical options:

```
plot(x=, y=, type="", col="", pch=, lty=, lwd=)
```

- Colors

```
colors()[1:20] # View first 20 elements of the color vector
#>  [1] "white"         "aliceblue"     "antiquewhite"  "antiquewhite1"
#>  [5] "antiquewhite2" "antiquewhite3" "antiquewhite4" "aquamarine"
#>  [9] "aquamarine1"   "aquamarine2"   "aquamarine3"   "aquamarine4"
#> [13] "azure"         "azure1"        "azure2"        "azure3"
#> [17] "azure4"        "beige"         "bisque"        "bisque1"
colors()[179] # View specific element of the color vector
#> [1] "gray26"
```

Another option: R Color Infographic

```
plot(x = gap$gdpPercap, y = gap$lifeExp, type="p", col=colors()[145]) # or col="gold3"
plot(x = gap$gdpPercap, y = gap$lifeExp, type="p", col="seagreen4") # or col=colors()[578]
```

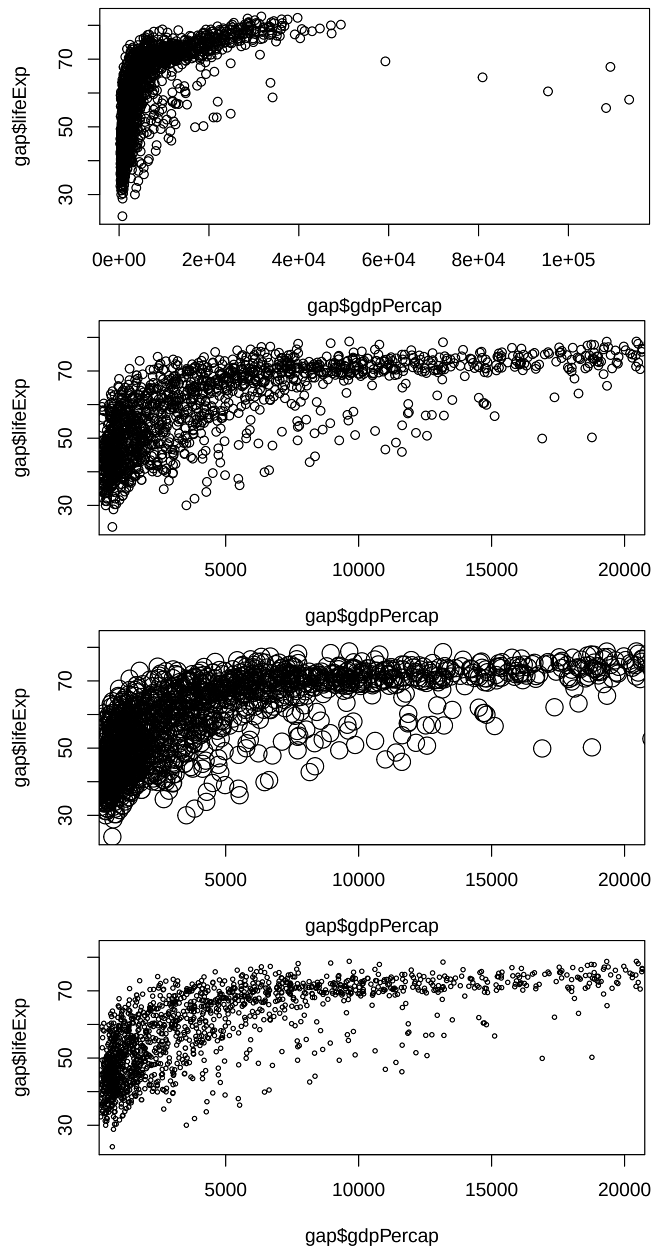

gap$lifeExp
30
50
70
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
gap$gdpPercap
5000
10000
15000
20000

Figure 13.6:

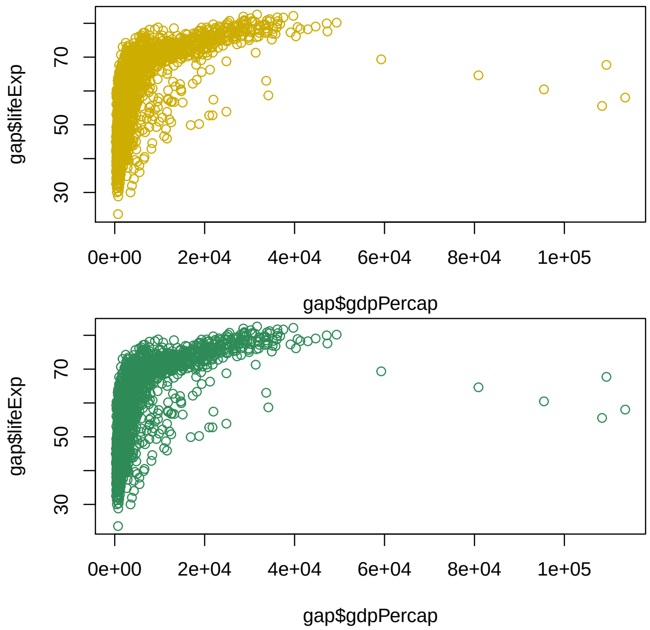


Figure 13.7:

- Point Styles and Widths

A Good Reference

```
# Change point style to crosses
plot(x = gap$gdpPercap, y = gap$lifeExp, type="p", pch=3)
# Change point style to filled squares
plot(x = gap$gdpPercap, y = gap$lifeExp, type="p",pch=15)
# Change point style to filled squares and increase point size to 3
plot(x = gap$gdpPercap, y = gap$lifeExp, type="p",pch=15, cex=3)
# Change point style to "w"
plot(x = gap$gdpPercap, y = gap$lifeExp, type="p", pch="w")
# Change point style to "$" and increase point size to 2
plot(x = gap$gdpPercap, y = gap$lifeExp, type="p",pch="$", cex=2)
```

- Line Styles and Widths

```
# Line plot with solid line
plot(x = gap$gdpPercap, y = gap$lifeExp, type="l", lty=1)
# Line plot with medium dashed line
plot(x = gap$gdpPercap, y = gap$lifeExp, type="l", lty=2)
# Line plot with short dashed line
plot(x = gap$gdpPercap, y = gap$lifeExp, type="l", lty=3)
# Change line width to 2
```

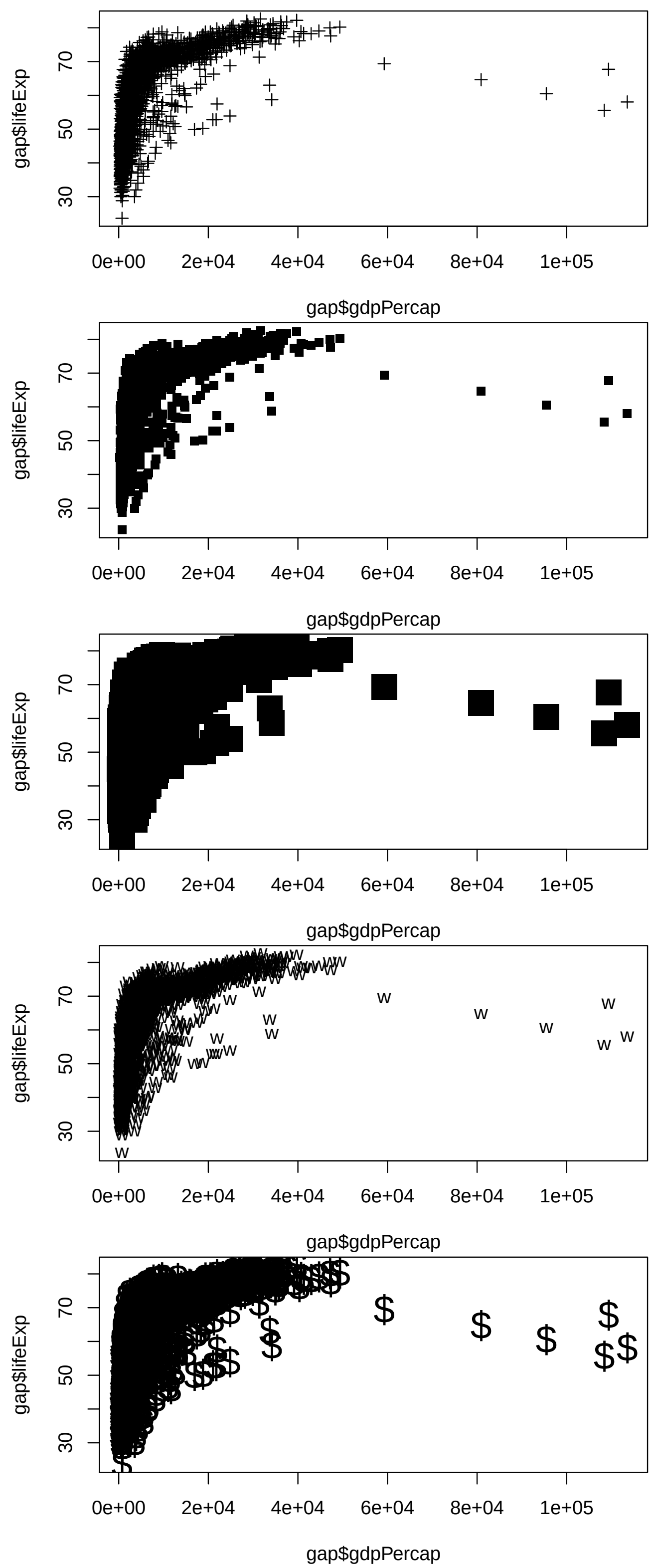

gap$lifeExp
gap$gdpPercap
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
30
50
70

Figure 13.8:

```
plot(x = gap$gdpPercap, y = gap$lifeExp, type="l", lty=3, lwd=2)
# Change line width to 5
plot(x = gap$gdpPercap, y = gap$lifeExp, type="l",  lwd=5)
# Change line width to 10 and use dash-dot
plot(x = gap$gdpPercap, y = gap$lifeExp, type="l",  lty=4, lwd=10)
```

13.2.6 Annotations, Reference Lines, and Legends

- Text

We can add text to an arbitrary point on the graph like this:

```
# plot the line first
plot(x = gap$gdpPercap, y = gap$lifeExp, type="p")
# now add the label
text(x=40000, y=50, labels="Evens Out", cex = .75)
```

We can also add labels for every point by passing in a vector of text:

```
# first randomly select rows for a smaller gapaset
library(dplyr)
small <- gap %>% sample_n(100)

# plot the line first
plot(x = small$gdpPercap, y = small$lifeExp, type="p")
# now add the label
text(x = small$gdpPercap, y = small$lifeExp, labels = small$country)
```

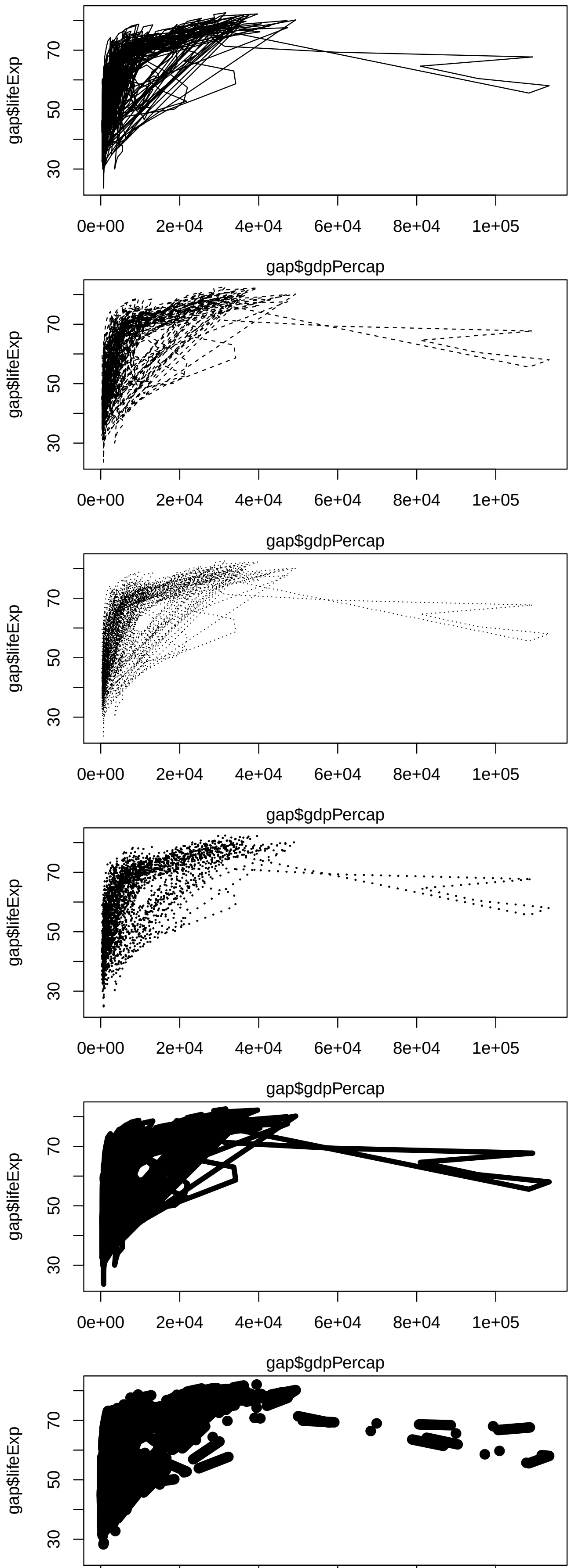
- Reference Lines

```
# plot the line
plot(x = gap$gdpPercap, y = gap$lifeExp, type="p")
# now the guides
abline(v=40000, h=75, lty=2)
```

13.3 ggplot2

Setup:

```
library(ggplot2)
gap <- read.csv("data/gapminder-FiveYearData.csv", stringsAsFactors = F)
```


gap$lifeExp
30
50
70
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
gap$gdpPercap
gap$lifeExp
30
50
70
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
gap$gdpPercap
gap$lifeExp
30
50
70
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
gap$gdpPercap
gap$lifeExp
30
50
70
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
gap$gdpPercap
gap$lifeExp
30
50
70
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
gap$gdpPercap
gap$lifeExp
30
50
70

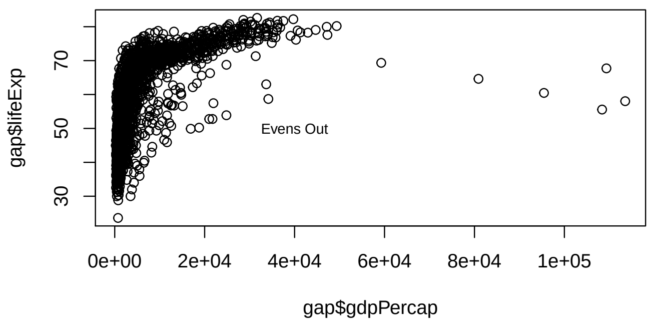


Figure 13.10:

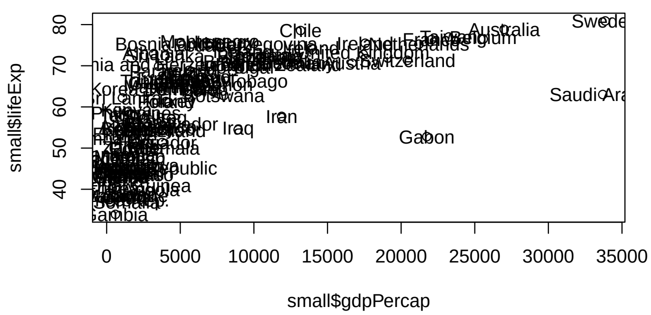


Figure 13.11:

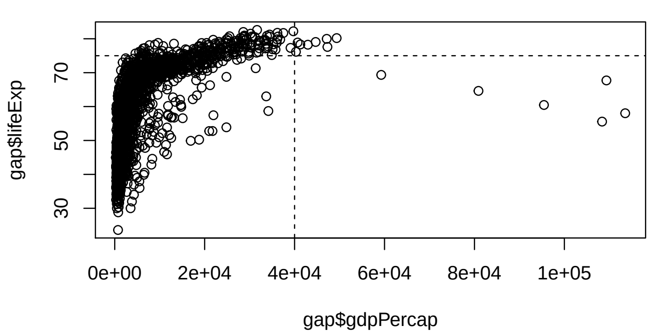


Figure 13.12:

Why `ggplot`?

- More elegant & compact code than R base graphics
- More aesthetically pleasing defaults than lattice
- Very powerful for exploratory data analysis
- Follows a grammar, just like any language.
- It defines basic components that make up a sentence. In this case, the grammar defines components in a plot.
- *G*rammar of *g*raphics originally coined by Lee Wilkinson

13.3.1 Grammar

The general call for `ggplot2` looks like this:

```
ggplot(data=, aes(x=, y=), color=, size=,) + geom_xxxx()+geom_yyyy()
```

The *grammar* involves some basic components:

1. **Data**: a data.frame
2. **Aes**thetics: How your data are represented visually, aka its "mapping". Which variables are shown on x, y axes, as well as color, size, shape, etc.
3. **Geom**etry: The geometric objects in a plot – points, lines, polygons, etc.

The key to understanding `ggplot2` is thinking about a figure in layers: just like you might do in an image editing program like Photoshop, Illustrator, or Inkscape.

Let's look at an example:

```
ggplot(data = gap, aes(x = gdpPercap, y = lifeExp)) +
  geom_point()
```

So the first thing we do is call the `ggplot` function. This function lets R know that we're creating a new plot, and any of the arguments we give the `ggplot` function are the global options for the plot: they apply to all layers on the plot.

We've passed in two arguments to `ggplot`. First, we tell `ggplot` what **`data`** we want to show on our figure, in this example the `gapminder` data we read in earlier.

For the second argument we passed in the **`aes`** function, which tells `ggplot` how variables in the data map to aesthetic properties of the figure, in this case the x and y locations. Here we told `ggplot` we want to plot the `lifeExp` column of the gapminder data frame on the x-axis, and the `gdpPercap` column on the y-axis.

Notice that we didn't need to explicitly pass `aes` these columns (e.g. `x = gapminder[, "lifeExp""])`, this is because `ggplot` is smart enough to know to look in the data for that column!

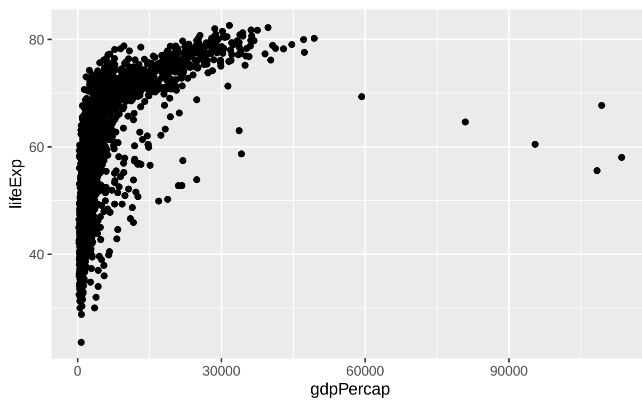
By itself, the call to `ggplot` isn't enough to draw a figure:

```
ggplot(data = gap, aes(x = gdpPercap, y = lifeExp))
```

We need to tell `ggplot` how we want to visually represent the data, which we do by adding a new **`geom`** layer. In our example, we used `geom_point`, which tells `ggplot` we want to visually represent the relationship between x and y as a scatterplot of points:

```
ggplot(data = gap, aes(x = gdpPercap, y = lifeExp)) +
  geom_point()

# same as
# my_plot <- ggplot(data = gap, aes(x = gdpPercap, y = lifeExp))
# my_plot + geom_point()
```



Challenge 1.

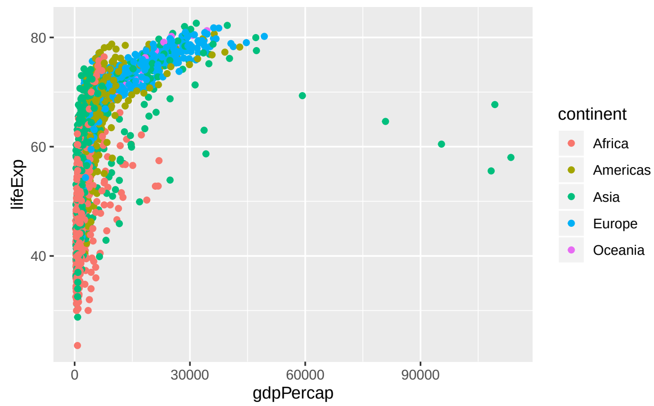
Modify the example so that the figure visualises how life expectancy has changed over time:

Hint: the gapminder dataset has a column called `year`, which should appear on the x-axis.

13.3.2 Anatomy of `aes`

In the previous examples and challenge we've used the `aes` function to tell the scatterplot **`geom`** about the **`x`** and **`y`** locations of each point. Another aesthetic property we can modify is the point **color**.

```
ggplot(data = gap, aes(x = gdpPercap, y = lifeExp, color=continent)) +
  geom_point()
```



Normally, specifying options like `color="red"` or `size=10` for a given layer results in its contents being red and quite large. Inside the `aes()` function, however, these arguments are given entire variables whose values will then be displayed using different realizations of that aesthetic.

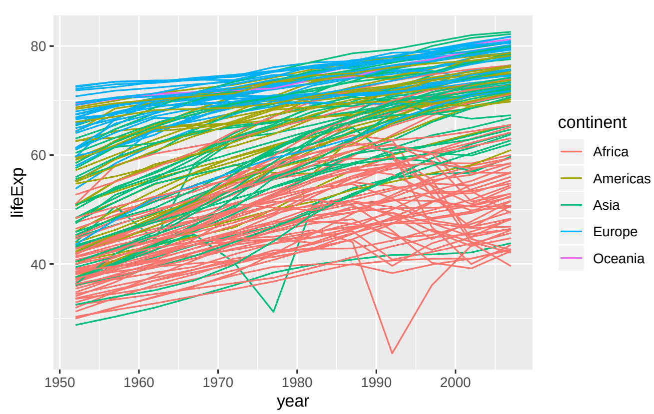
Color isn't the only aesthetic argument we can set to display variation in the data. We can also vary by shape, size, etc.

```
ggplot(data=, aes(x=, y=, by =, color=, linetype=, shape=, size=))
```

13.3.3 Layers

In the previous challenge, you plotted `lifExp` over time. Using a scatterplot probably isn't the best for visualising change over time. Instead, let's tell `ggplot` to visualise the data as a line plot:

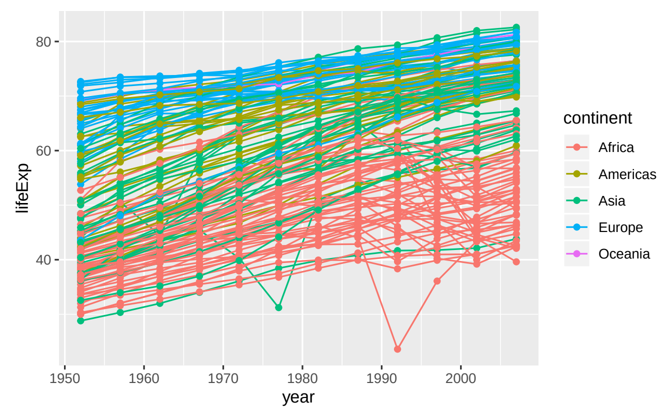
```
ggplot(data = gap, aes(x=year, y=lifeExp, by=country, color=continent)) +
  geom_line()
```



Instead of adding a `geom_point` layer, we've added a `geom_line` layer. We've also added the **by** aesthetic, which tells `ggplot` to draw a line for each country.

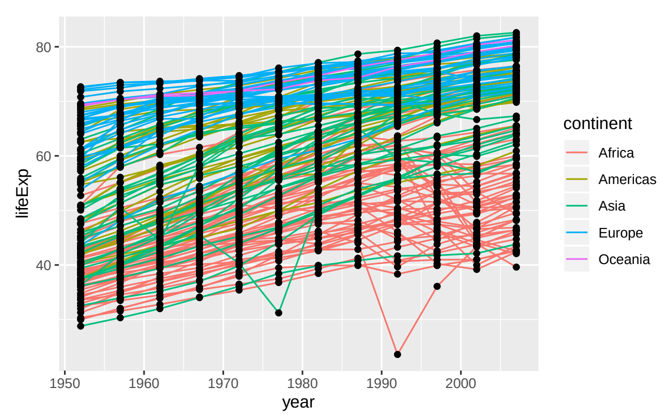
But what if we want to visualise both lines and points on the plot? We can simply add another layer to the plot:

```
ggplot(data = gap, aes(x=year, y=lifeExp, by=country, color=continent)) +
  geom_line() +
  geom_point()
```



It's important to note that each layer is drawn on top of the previous layer. In this example, the points have been drawn on top of the lines. Here's a demonstration:

```
ggplot(data = gap, aes(x=year, y=lifeExp, by=country)) +
  geom_line(aes(color=continent)) +
  geom_point()
```



In this example, the aesthetic mapping of **color** has been moved from the global plot options in `ggplot` to the `geom_line` layer so it no longer applies to the points. Now we can clearly see that the points are drawn on top of the lines.

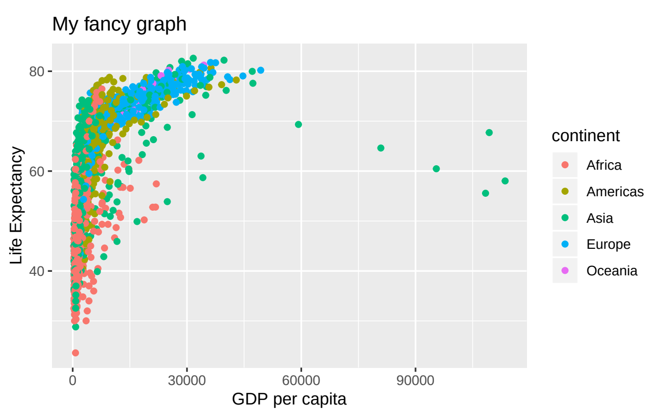
Challenge 2.

Switch the order of the point and line layers from the previous example. What happened?

13.3.4 Labels

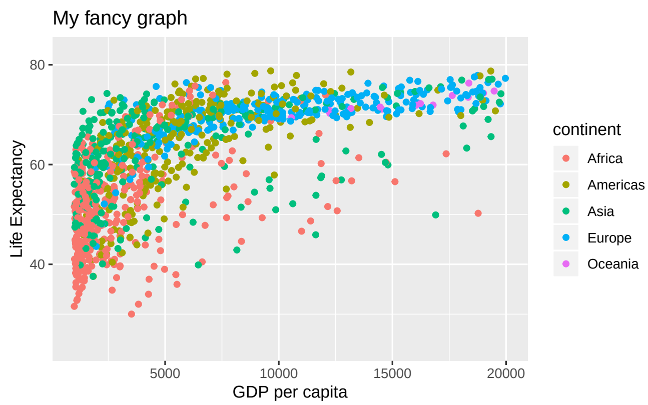
Labels are considered to be their own layers in `ggplot`.

```
# add x and y axis labels
ggplot(data = gap, aes(x = gdpPercap, y = lifeExp, color=continent)) +
  geom_point() +
  xlab("GDP per capita") +
  ylab("Life Expectancy") +
  ggtitle("My fancy graph")
```



So are scales:

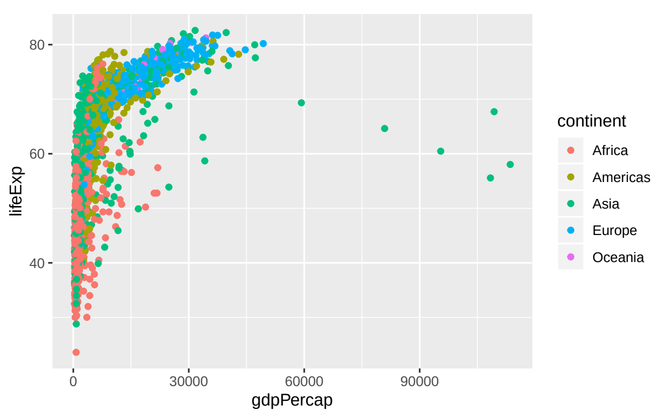
```
# limit x-axis from 1,000 to 20,000
ggplot(data = gap, aes(x = gdpPercap, y = lifeExp, color=continent)) +
  geom_point() +
  xlab("GDP per capita") +
  ylab("Life Expectancy") +
  ggtitle("My fancy graph") +
  xlim(1000, 20000)
#> Warning: Removed 515 rows containing missing values (geom_point).
```



13.3.5 Transformations and Stats

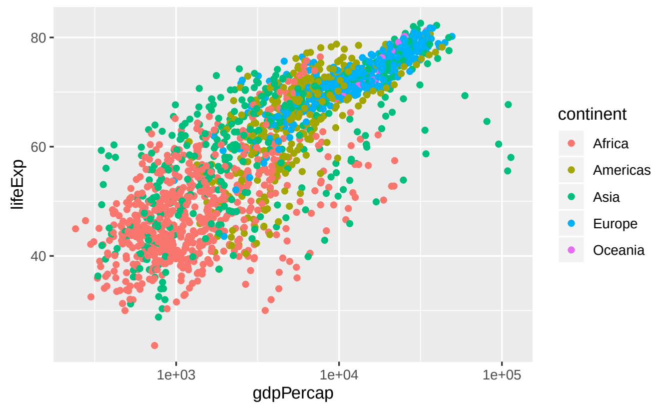
ggplot also makes it easy to overlay statistical models over the data. To demonstrate we'll go back to an earlier example:

```
ggplot(data = gap, aes(x = gdpPercap, y = lifeExp, color=continent)) +
  geom_point()
```



We can change the scale of units on the x-axis using the scale functions. These control the mapping between the data values and visual values of an aesthetic.

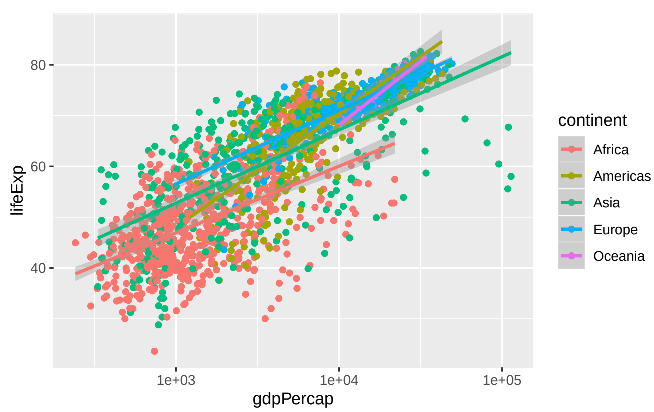
```
ggplot(data = gap, aes(x = gdpPercap, y = lifeExp, color=continent)) +
  geom_point() +
  scale_x_log10()
```



The `log10` function applied a transformation to the values of the `gdpPercap` column before rendering them on the plot, so that each multiple of 10 now only corresponds to an increase in 1 on the transformed scale, e.g. a GDP per capita of 1,000 is now 3 on the y axis, a value of 10,000 corresponds to 4 on the x-axis and so on. This makes it easier to visualise the spread of data on the x-axis.

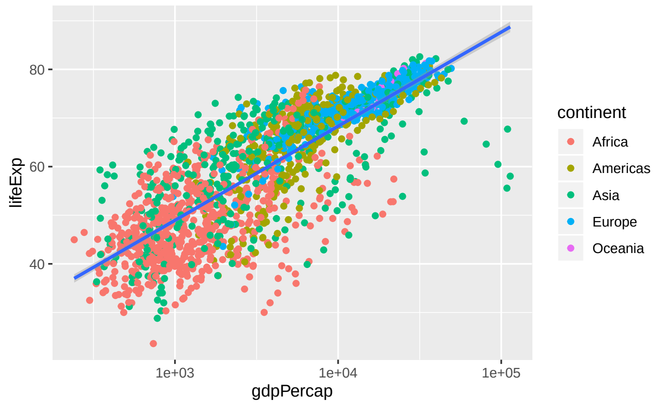
We can fit a simple relationship to the data by adding another layer, `geom_smooth`:

```
ggplot(data = gap, aes(x = gdpPercap, y = lifeExp, color=continent)) +
  geom_point() +
  scale_x_log10() +
  geom_smooth(method="lm")
```



Note that we have 5 lines, one for each region, because of the **color** option is the global `aes` function. But if we move it, we get different results:

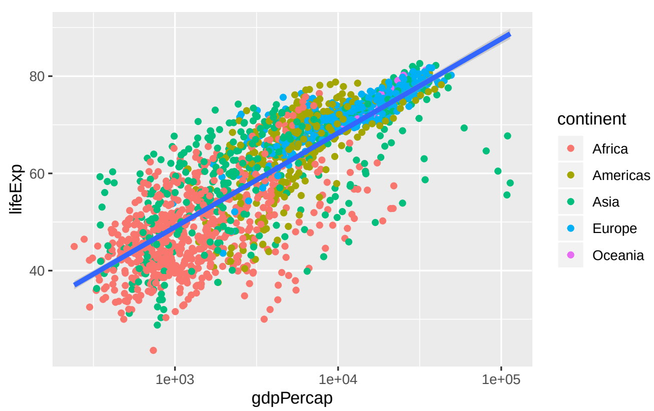
```
ggplot(data = gap, aes(x = gdpPercap, y = lifeExp)) +
  geom_point(aes(color=continent)) +
  scale_x_log10() +
  geom_smooth(method="lm")
```



So, there are two ways an aesthetic can be specified. Here, we set the **color** aesthetic by passing it as an argument to `geom_point`. Previously in the lesson, we used the `aes` function to define a *mapping* between data variables and their visual representation.

We can make the line thicker by setting the **size** aesthetic in the geom_smooth layer:

```
ggplot(data = gap, aes(x = gdpPercap, y = lifeExp)) +
  geom_point(aes(color=continent)) +
  scale_x_log10() +
  geom_smooth(method="lm", size = 1.5)
```



Challenge 3.

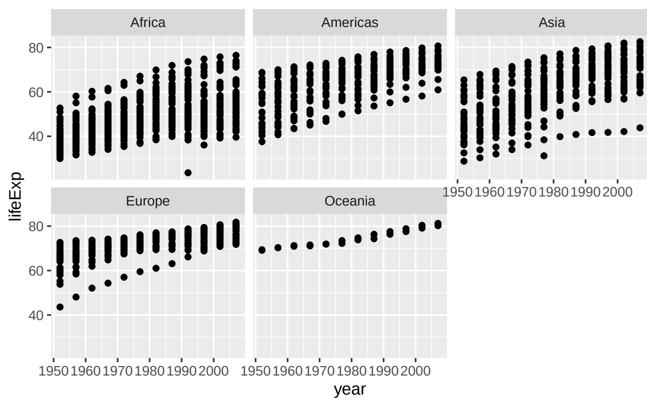
Modify the color and size of the points on the point layer in the previous example so that they are fixed (i.e. not reflective of continent).

Hint: do not use the `aes` function.

13.3.6 Facets

Earlier, we visualised the change in life expectancy over time across all countries in one plot. Alternatively, we can split this out over multiple panels by adding a layer of **facet** panels:

```
ggplot(data = gap, aes(x = year, y = lifeExp)) +
  geom_point() +
  facet_wrap( ~ continent)
```



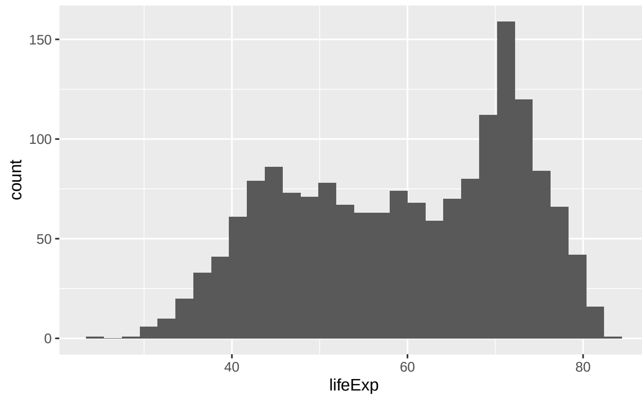
13.3.7 Putting Everything Together

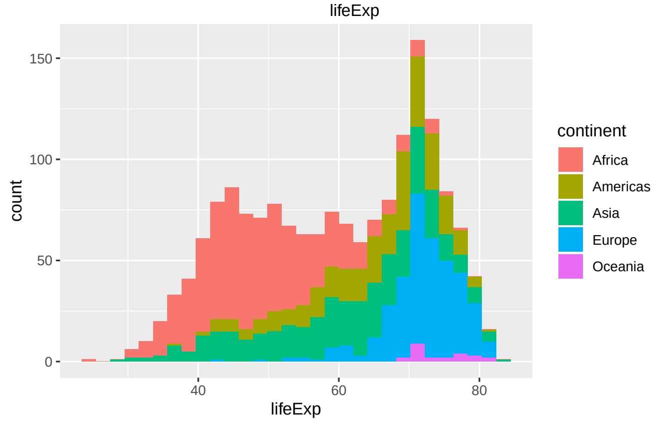
Here are some other common `geom` layers:

bar plots

```
# count of lifeExp bins
ggplot(data = gap, aes(x = lifeExp)) +
  geom_bar(stat="bin")
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

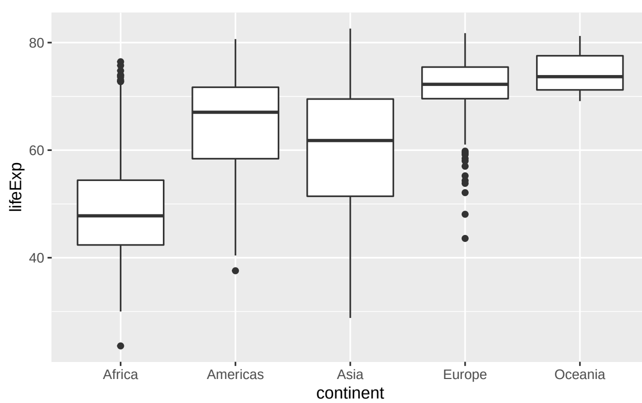
# with color representing regions
ggplot(data = gap, aes(x = lifeExp, fill = continent)) +
  geom_bar(stat="bin")
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





box plots

```
ggplot(data = gap, aes(x = continent, y = lifeExp)) +
  geom_boxplot()
```



This is just a taste of what you can do with `ggplot2`.

RStudio provides a really useful cheat sheet of the different layers available, and more extensive documentation is available on the ggplot2 website.

Finally, if you have no idea how to change something, a quick Google search will usually send you to a relevant question and answer on Stack Overflow with reusable code to modify!

Challenge 4.

Create a density plot of GDP per capita, filled by continent.

Advanced: - Transform the x-axis to better visualise the data spread. - Add a facet layer to panel the density plots by year.

13.4 Saving plots

There are two basic image types:

1) **Raster/Bitmap** (.png, .jpeg)

Every pixel of a plot contains its own separate coding; not so great if you want to resize the image.

```
jpeg(filename="example.png", width=, height=)
plot(x,y)
dev.off()
```

2) **Vector** (.pdf, .ps)

Every element of a plot is encoded with a function that gives its coding conditional on several factors; this is great for resizing.

```
pdf(filename="example.pdf", width=, height=)
plot(x,y)
dev.off()
```

Exporting with ggplot

```
# Assume we saved our plot is an object called example.plot
ggsave(filename="example.pdf", plot=example.plot, scale=, width=, height=)
```

Chapter 14

Statistical Inferences

```r
# setup
gap <- read.csv("data/gapminder-FiveYearData.csv", stringsAsFactors = TRUE)
```

14.1 Statistical Distributions

Since R was developed by statisticians, it handles distributions and simulation seamlessly.

All commonly-used distributions have functions in R. Each distribution has a family of functions:

- `d` - probability density/mass function, e.g. `dnorm()`
- `r` - generate a random value, e.g., `rnorm()`
- `p` - cumulative distribution function, e.g., `pnorm()`
- `q` - quantile function (inverse CDF), e.g., `qnorm()`

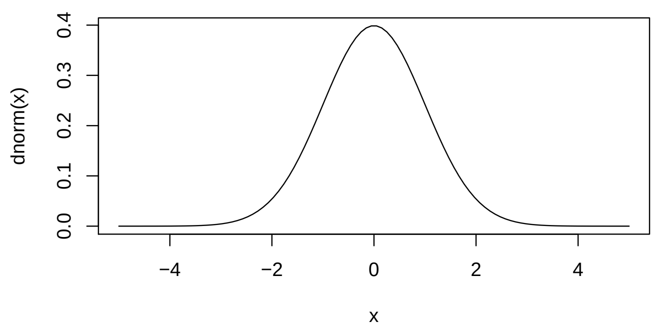
Let's see some of these functions in action with the normal distribution (mean 0, standard deviation 1)

```r
dnorm(1.96) # probability density of 1.96 from normal distribution
#> [1] 0.0584
rnorm(1:10) # get 10 random values from the normal distribution
#>  [1] -1.40004  0.25532 -2.43726 -0.00557  0.62155  1.14841 -1.82182
#>  [8] -0.24733 -0.24420 -0.28271
pnorm(1.96) # cumultative distribution function
#> [1] 0.975
qnorm(.975) # inverse cumulative distribution function
#> [1] 1.96
```

We can also use these functions on other distributions:

- `rnorm()` # normal distribution
- `runif()` # uniform distribution
- `rbinom()` # binomial distribution
- `rpois()` # poisson distribtion
- `rbeta()` # beta distribution
- `rgamma()` # gamma distribution
- `rt()` # student t distribution
- `rchisq()`. # chi-squared distribution

```
rbinom(0:10, size = 10, prob = 0.3)
#>  [1] 2 4 4 2 6 4 1 3 4 4 1
dt(5, df = 1)
#> [1] 0.0122

x <- seq(-5, 5, length = 100)
plot(x, dnorm(x), type = 'l')
```



14.1.1 Sampling and Simulation

We can draw a sample with or without replacement with `sample`.

```
sample(1:nrow(gap), 20, replace = FALSE)
#>  [1]  447 1284  752  674 1699 1241  726 1579 1568 1094  265  150 1023 1290
#> [15] 1440  117  930 1553  180 1208
```

`dplyr` has a helpful `select_n` function that samples rows of a dataframe.

```
small <- sample_n(gap, 20)
nrow(small)
#> [1] 20
```

Here's an example of some code that would be part of a bootstrap.

```
gap <- read.csv("data/gapminder-FiveYearData.csv", stringsAsFactors = F)
```

```
# actual mean
mean(gap$lifeExp, na.rm = TRUE)
#> [1] 59.5

# here's a bootstrap sample:
smp <- sample_n(gap, size = nrow(gap), replace = TRUE)
mean(smp$lifeExp, na.rm = TRUE)
#> [1] 59.4
```

14.1.2 Random Seeds

A few key facts about generating random numbers:

- Random numbers on a computer are *pseudo-random*; they are generated deterministically from a very, very, very long sequence that repeats
- The *seed* determines where you are in that sequence

To replicate any work involving random numbers, make sure to set the seed first. The seed can be arbitrary – pick your favorite number.

```
set.seed(1)
vals <- sample(1:nrow(gap), 10)
vals
#>  [1] 1017  679  129  930 1533  471  299  270 1211 1331

vals <- sample(1:nrow(gap), 10)
vals
#>  [1]  597 1301 1518  330 1615   37 1129  729  878  485

set.seed(1)
vals <- sample(1:nrow(gap), 10)
vals
#>  [1] 1017  679  129  930 1533  471  299  270 1211 1331
```

14.1.3 Challenges

Challenge 1.

Generate 100 random Poisson values with a population mean of 5. How close is the mean of those 100 values to the value of 5?

Challenge 2.

What is the 95th percentile of a chi-square distribution with 1 degree of freedom?

Challenge 3.

What's the probability of getting a value greater than 5 if you draw from a standard normal distribution? What about a t distribution with 1 degree of freedom?

14.2 Inferences and Regressions

Once we've imported our data, summarized it, carried out group-wise operations, and perhaps reshaped it, we may also want to attempt causal inference.

This often requires doing the following:

1) Carrying out Classical Hypothesis Tests
2) Estimating Regressions

```
# setup
gap <- read.csv("data/gapminder-FiveYearData.csv", stringsAsFactors = F)
```

14.2.1 Statistical Tests

Let's say we're interested in whether the life expectancy in 1967 is different than in 1977.

```
# pull out life expectancy by different years
life.exp.1967 <- gap$lifeExp[gap$year==1967]
life.exp.1977 <- gap$lifeExp[gap$year==1977]
```

One can test for differences in distributions in either:

1) Their means using t-tests:

```
# t test of means
t.test(x = life.exp.1967, y = life.exp.1977)
#>
#>  Welch Two Sample t-test
#>
#> data:  life.exp.1967 and life.exp.1977
#> t = -3, df = 281, p-value = 0.005
#> alternative hypothesis: true difference in means is not equal to 0
#> 95 percent confidence interval:
#>  -6.57 -1.21
#> sample estimates:
#> mean of x mean of y
#>      55.7      59.6
```

2) Their entire distributions using ks-tests

```r
# ks tests of distributions
ks.test(x = life.exp.1967, y = life.exp.1977)
#> Warning in ks.test(x = life.exp.1967, y = life.exp.1977): p-value will be
#> approximate in the presence of ties
#>
#>  Two-sample Kolmogorov-Smirnov test
#>
#> data:  life.exp.1967 and life.exp.1977
#> D = 0.2, p-value = 0.008
#> alternative hypothesis: two-sided
```

14.2.2 Regressions and Linear Models

Running regressions in R is generally straightforward. There are two basic, catch-all regression functions in R:

- *glm* fits a generalized linear model with your choice of family/link function (gaussian, logit, poisson, etc.)
- *lm* is just a standard linear regression (equivalent to glm with family = gaussian(link = "identity"))

The basic `glm` call looks something like this:

```r
glm(formula = y ~ x1 + x2 + x3 + ..., family = familyname(link = "linkname"), data = )
```

There are a bunch of families and links to use (`?family` for a full list), but some essentials are: **binomial(link = "logit")**, **gaussian(link = "identity")**, and **poisson(link = "log")**

If you're using `lm`, the call looks the same but without the `family` argument.

- Example: suppose we want to regress the life expectency on the GDP per capita and the population, as well as the continent and year. The `lm` call would be something like this:

```r
reg <- lm(formula = lifeExp ~ log(gdpPercap) + log(pop) + continent + year, data = gap)
```

Missing values

Missing values obviously cannot convey any information about the relationship between the variables. Most modeling functions will drop any rows that contain missing values.

14.2.3 Regression Output

When we store this regression in an object, we get access to several items of interest.

1. All components contained in the regression output:

```
names(reg)
#>  [1] "coefficients" "residuals"     "effects"       "rank"
#>  [5] "fitted.values" "assign"        "qr"            "df.residual"
#>  [9] "contrasts"     "xlevels"       "call"          "terms"
#> [13] "model"
```

2. Regression coefficients

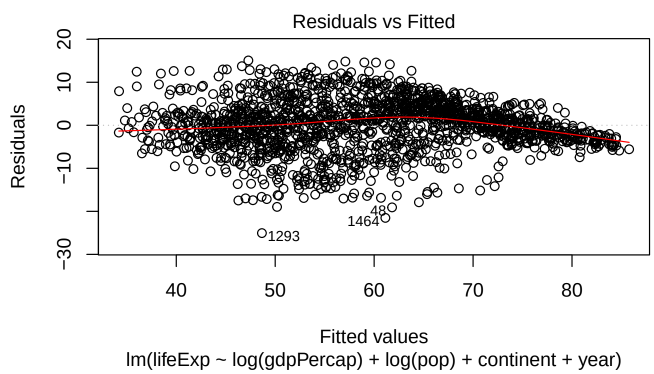
```
reg$coefficients
#>       (Intercept)    log(gdpPercap)          log(pop) continentAmericas
#>          -460.813             5.076             0.153             8.745
#>     continentAsia   continentEurope  continentOceania              year
#>             6.825            12.281            12.540             0.238
```

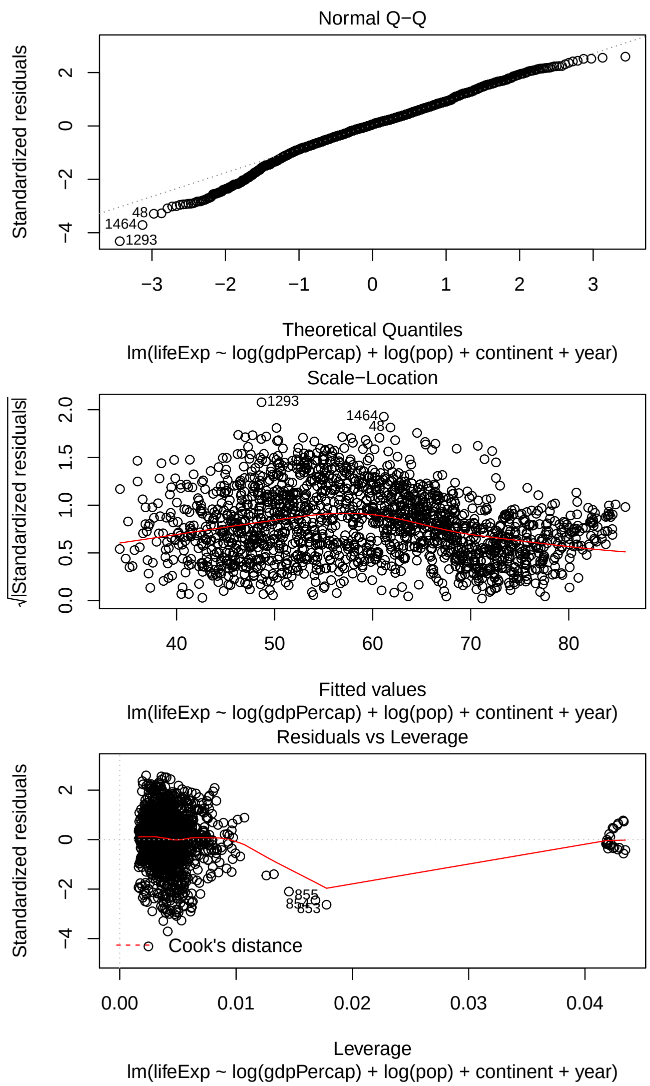
3. Regression degrees of freedom

```
reg$df.residual
#> [1] 1696
```

4. Standard (diagnostic) plots for a regression

```
plot(reg)
```





R also has a helpful `summary` method for regression objects.

```
summary(reg)
#> 
#> Call:
#> lm(formula = lifeExp ~ log(gdpPercap) + log(pop) + continent + 
#>     year, data = gap)
#> 
#> Residuals:
#>     Min      1Q  Median      3Q     Max 
#> -25.057  -3.286   0.329   3.706  15.065 
#> 
```

```
#> Coefficients:
#>                     Estimate Std. Error t value Pr(>|t|)
#> (Intercept)        -4.61e+02   1.70e+01  -27.15   <2e-16 ***
#> log(gdpPercap)      5.08e+00   1.63e-01   31.19   <2e-16 ***
#> log(pop)            1.53e-01   9.67e-02    1.58     0.11
#> continentAmericas   8.75e+00   4.77e-01   18.35   <2e-16 ***
#> continentAsia       6.83e+00   4.23e-01   16.13   <2e-16 ***
#> continentEurope     1.23e+01   5.29e-01   23.20   <2e-16 ***
#> continentOceania    1.25e+01   1.28e+00    9.79   <2e-16 ***
#> year                2.38e-01   8.93e-03   26.61   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 5.81 on 1696 degrees of freedom
#> Multiple R-squared:  0.798,  Adjusted R-squared:  0.798
#> F-statistic:  960 on 7 and 1696 DF,  p-value: <2e-16
```

We can also extract useful things from the summary object:

```
# Store summary method results
summ_reg <- summary(reg)

# View summary method results objects
objects(summ_reg)
#>  [1] "adj.r.squared" "aliased"       "call"          "coefficients"
#>  [5] "cov.unscaled"  "df"            "fstatistic"    "r.squared"
#>  [9] "residuals"     "sigma"         "terms"

# View table of coefficients
summ_reg$coefficients
#>                   Estimate Std. Error t value  Pr(>|t|)
#> (Intercept)       -460.813   16.97028  -27.15 3.96e-135
#> log(gdpPercap)       5.076    0.16272   31.19 3.37e-169
#> log(pop)             0.153    0.09668    1.58  1.14e-01
#> continentAmericas    8.745    0.47660   18.35  9.61e-69
#> continentAsia        6.825    0.42320   16.13  1.49e-54
#> continentEurope     12.281    0.52924   23.20 1.12e-103
#> continentOceania    12.540    1.28114    9.79  4.80e-22
#> year                 0.238    0.00893   26.61 1.06e-130
```

14.2.4 Interactions

There are also some useful shortcuts for regressing on interaction terms:

1. `x1:x2` interacts all terms in x1 with all terms in x2

```
mod.1 <- lm(lifeExp ~ log(gdpPercap) + log(pop) + continent:factor(year), data = gap)
summary(mod.1)
#> 
#> Call:
#> lm(formula = lifeExp ~ log(gdpPercap) + log(pop) + continent:factor(year),
#>     data = gap)
#> 
#> Residuals:
#>     Min      1Q  Median      3Q     Max 
#> -26.568  -2.553   0.004   2.915  15.567 
#> 
#> Coefficients: (1 not defined because of singularities)
#>                                    Estimate Std. Error t value Pr(>|t|)
#> (Intercept)                         27.1838     4.6849    5.80  7.8e-09
#> log(gdpPercap)                       5.0795     0.1605   31.65  < 2e-16
#> log(pop)                             0.0789     0.0943    0.84  0.40251
#> continentAfrica:factor(year)1952   -24.1425     4.1125   -5.87  5.2e-09
#> continentAmericas:factor(year)1952 -16.4465     4.1663   -3.95  8.2e-05
#> continentAsia:factor(year)1952     -19.3347     4.1408   -4.67  3.3e-06
#> continentEurope:factor(year)1952    -7.0918     4.1352   -1.71  0.08654
#> continentOceania:factor(year)1952   -6.0635     5.6511   -1.07  0.28344
#> continentAfrica:factor(year)1957   -22.4964     4.1098   -5.47  5.1e-08
#> continentAmericas:factor(year)1957 -14.3673     4.1643   -3.45  0.00057
#> continentAsia:factor(year)1957     -17.1743     4.1375   -4.15  3.5e-05
#> continentEurope:factor(year)1957    -5.9094     4.1327   -1.43  0.15293
#> continentOceania:factor(year)1957   -5.6300     5.6503   -1.00  0.31921
#> continentAfrica:factor(year)1962   -21.0139     4.1069   -5.12  3.5e-07
#> continentAmericas:factor(year)1962 -12.3135     4.1630   -2.96  0.00314
#> continentAsia:factor(year)1962     -15.5626     4.1351   -3.76  0.00017
#> continentEurope:factor(year)1962    -5.0542     4.1308   -1.22  0.22130
#> continentOceania:factor(year)1962   -5.3122     5.6498   -0.94  0.34723
#> continentAfrica:factor(year)1967   -19.7034     4.1035   -4.80  1.7e-06
#> continentAmericas:factor(year)1967 -10.9324     4.1613   -2.63  0.00869
#> continentAsia:factor(year)1967     -13.1569     4.1327   -3.18  0.00148
#> continentEurope:factor(year)1967    -4.9134     4.1291   -1.19  0.23423
#> continentOceania:factor(year)1967   -5.7712     5.6492   -1.02  0.30712
#> continentAfrica:factor(year)1972   -18.1469     4.1007   -4.43  1.0e-05
#> continentAmericas:factor(year)1972  -9.6537     4.1595   -2.32  0.02042
#> continentAsia:factor(year)1972     -11.6014     4.1293   -2.81  0.00502
#> continentEurope:factor(year)1972    -4.9763     4.1275   -1.21  0.22813
#> continentOceania:factor(year)1972   -5.8094     5.6487   -1.03  0.30389
#> continentAfrica:factor(year)1977   -16.1848     4.0996   -3.95  8.2e-05
#> continentAmericas:factor(year)1977  -8.3382     4.1580   -2.01  0.04509
#> continentAsia:factor(year)1977     -10.1220     4.1270   -2.45  0.01428
#> continentEurope:factor(year)1977    -4.5523     4.1267   -1.10  0.27013
```

```
#> continentOceania:factor(year)1977   -5.1232     5.6485  -0.91 0.36454
#> continentAfrica:factor(year)1982   -14.1933     4.0990  -3.46 0.00055
#> continentAmericas:factor(year)1982  -6.5921     4.1577  -1.59 0.11304
#> continentAsia:factor(year)1982      -7.6001     4.1257  -1.84 0.06564
#> continentEurope:factor(year)1982    -4.1185     4.1262  -1.00 0.31837
#> continentOceania:factor(year)1982   -4.0553     5.6483  -0.72 0.47288
#> continentAfrica:factor(year)1987   -12.1850     4.0995  -2.97 0.00300
#> continentAmericas:factor(year)1987  -4.7157     4.1577  -1.13 0.25687
#> continentAsia:factor(year)1987      -5.6914     4.1249  -1.38 0.16785
#> continentEurope:factor(year)1987    -3.7298     4.1258  -0.90 0.36613
#> continentOceania:factor(year)1987   -3.5164     5.6480  -0.62 0.53364
#> continentAfrica:factor(year)1992   -11.8028     4.0994  -2.88 0.00404
#> continentAmericas:factor(year)1992  -3.2855     4.1575  -0.79 0.42949
#> continentAsia:factor(year)1992      -4.3823     4.1241  -1.06 0.28811
#> continentEurope:factor(year)1992    -2.5151     4.1262  -0.61 0.54225
#> continentOceania:factor(year)1992   -1.9804     5.6480  -0.35 0.72590
#> continentAfrica:factor(year)1997   -11.9577     4.0986  -2.92 0.00358
#> continentAmericas:factor(year)1997  -2.1611     4.1566  -0.52 0.60319
#> continentAsia:factor(year)1997      -3.5016     4.1228  -0.85 0.39583
#> continentEurope:factor(year)1997    -2.0843     4.1256  -0.51 0.61348
#> continentOceania:factor(year)1997   -1.4478     5.6478  -0.26 0.79771
#> continentAfrica:factor(year)2002   -12.5237     4.0972  -3.06 0.00227
#> continentAmericas:factor(year)2002  -0.9898     4.1564  -0.24 0.81180
#> continentAsia:factor(year)2002      -2.6798     4.1221  -0.65 0.51571
#> continentEurope:factor(year)2002    -1.5734     4.1252  -0.38 0.70294
#> continentOceania:factor(year)2002   -0.4735     5.6477  -0.08 0.93320
#> continentAfrica:factor(year)2007   -11.6568     4.0948  -2.85 0.00447
#> continentAmericas:factor(year)2007  -0.6931     4.1550  -0.17 0.86754
#> continentAsia:factor(year)2007      -2.2008     4.1202  -0.53 0.59332
#> continentEurope:factor(year)2007    -1.5284     4.1247  -0.37 0.71102
#> continentOceania:factor(year)2007        NA         NA     NA      NA
#>
#> (Intercept)                        ***
#> log(gdpPercap)                     ***
#> log(pop)
#> continentAfrica:factor(year)1952   ***
#> continentAmericas:factor(year)1952 ***
#> continentAsia:factor(year)1952     ***
#> continentEurope:factor(year)1952   .
#> continentOceania:factor(year)1952
#> continentAfrica:factor(year)1957   ***
#> continentAmericas:factor(year)1957 ***
#> continentAsia:factor(year)1957     ***
#> continentEurope:factor(year)1957
#> continentOceania:factor(year)1957
```

```
#> continentAfrica:factor(year)1962   ***
#> continentAmericas:factor(year)1962 **
#> continentAsia:factor(year)1962     ***
#> continentEurope:factor(year)1962
#> continentOceania:factor(year)1962
#> continentAfrica:factor(year)1967   ***
#> continentAmericas:factor(year)1967 **
#> continentAsia:factor(year)1967     **
#> continentEurope:factor(year)1967
#> continentOceania:factor(year)1967
#> continentAfrica:factor(year)1972   ***
#> continentAmericas:factor(year)1972 *
#> continentAsia:factor(year)1972     **
#> continentEurope:factor(year)1972
#> continentOceania:factor(year)1972
#> continentAfrica:factor(year)1977   ***
#> continentAmericas:factor(year)1977 *
#> continentAsia:factor(year)1977     *
#> continentEurope:factor(year)1977
#> continentOceania:factor(year)1977
#> continentAfrica:factor(year)1982   ***
#> continentAmericas:factor(year)1982
#> continentAsia:factor(year)1982     .
#> continentEurope:factor(year)1982
#> continentOceania:factor(year)1982
#> continentAfrica:factor(year)1987   **
#> continentAmericas:factor(year)1987
#> continentAsia:factor(year)1987
#> continentEurope:factor(year)1987
#> continentOceania:factor(year)1987
#> continentAfrica:factor(year)1992   **
#> continentAmericas:factor(year)1992
#> continentAsia:factor(year)1992
#> continentEurope:factor(year)1992
#> continentOceania:factor(year)1992
#> continentAfrica:factor(year)1997   **
#> continentAmericas:factor(year)1997
#> continentAsia:factor(year)1997
#> continentEurope:factor(year)1997
#> continentOceania:factor(year)1997
#> continentAfrica:factor(year)2002   **
#> continentAmericas:factor(year)2002
#> continentAsia:factor(year)2002
#> continentEurope:factor(year)2002
#> continentOceania:factor(year)2002
```

```
#> continentAfrica:factor(year)2007   **
#> continentAmericas:factor(year)2007
#> continentAsia:factor(year)2007
#> continentEurope:factor(year)2007
#> continentOceania:factor(year)2007
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 5.65 on 1642 degrees of freedom
#> Multiple R-squared:  0.816,  Adjusted R-squared:  0.809
#> F-statistic:  119 on 61 and 1642 DF,  p-value: <2e-16
```

2. `x1*x2` produces the cross of x1 and x2, or x1+x2+x1:x2

```
mod.2 <- lm(lifeExp ~ log(gdpPercap) + log(pop) + continent*factor(year), data = gap)
summary(mod.2)
#>
#> Call:
#> lm(formula = lifeExp ~ log(gdpPercap) + log(pop) + continent *
#>     factor(year), data = gap)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -26.568  -2.553   0.004   2.915  15.567
#>
#> Coefficients:
#>                                    Estimate Std. Error t value Pr(>|t|)
#> (Intercept)                          3.0413     2.0741    1.47  0.14275
#> log(gdpPercap)                       5.0795     0.1605   31.65  < 2e-16
#> log(pop)                             0.0789     0.0943    0.84  0.40251
#> continentAmericas                    7.6960     1.3932    5.52  3.9e-08
#> continentAsia                        4.8078     1.2657    3.80  0.00015
#> continentEurope                     17.0508     1.3295   12.83  < 2e-16
#> continentOceania                    18.0790     4.0890    4.42  1.0e-05
#> factor(year)1957                     1.6461     1.1078    1.49  0.13747
#> factor(year)1962                     3.1286     1.1084    2.82  0.00482
#> factor(year)1967                     4.4392     1.1097    4.00  6.6e-05
#> factor(year)1972                     5.9956     1.1113    5.39  7.9e-08
#> factor(year)1977                     7.9578     1.1124    7.15  1.3e-12
#> factor(year)1982                     9.9492     1.1134    8.94  < 2e-16
#> factor(year)1987                    11.9575     1.1138   10.74  < 2e-16
#> factor(year)1992                    12.3398     1.1146   11.07  < 2e-16
#> factor(year)1997                    12.1848     1.1161   10.92  < 2e-16
#> factor(year)2002                    11.6188     1.1181   10.39  < 2e-16
#> factor(year)2007                    12.4857     1.1212   11.14  < 2e-16
#> continentAmericas:factor(year)1957   0.4330     1.9438    0.22  0.82375
```

```
#> continentAsia:factor(year)1957       0.5142     1.7776    0.29  0.77241
#> continentEurope:factor(year)1957    -0.4638     1.8313   -0.25  0.80010
#> continentOceania:factor(year)1957   -1.2126     5.7552   -0.21  0.83315
#> continentAmericas:factor(year)1962   1.0043     1.9438    0.52  0.60546
#> continentAsia:factor(year)1962       0.6435     1.7777    0.36  0.71741
#> continentEurope:factor(year)1962    -1.0911     1.8315   -0.60  0.55142
#> continentOceania:factor(year)1962   -2.3774     5.7552   -0.41  0.67960
#> continentAmericas:factor(year)1967   1.0750     1.9438    0.55  0.58033
#> continentAsia:factor(year)1967       1.7387     1.7777    0.98  0.32819
#> continentEurope:factor(year)1967    -2.2608     1.8317   -1.23  0.21728
#> continentOceania:factor(year)1967   -4.1468     5.7552   -0.72  0.47130
#> continentAmericas:factor(year)1972   0.7972     1.9438    0.41  0.68176
#> continentAsia:factor(year)1972       1.7377     1.7779    0.98  0.32851
#> continentEurope:factor(year)1972    -3.8801     1.8322   -2.12  0.03435
#> continentOceania:factor(year)1972   -5.7415     5.7552   -1.00  0.31862
#> continentAmericas:factor(year)1977   0.1505     1.9439    0.08  0.93828
#> continentAsia:factor(year)1977       1.2549     1.7784    0.71  0.48050
#> continentEurope:factor(year)1977    -5.4183     1.8329   -2.96  0.00316
#> continentOceania:factor(year)1977   -7.0175     5.7553   -1.22  0.22290
#> continentAmericas:factor(year)1982  -0.0948     1.9439   -0.05  0.96110
#> continentAsia:factor(year)1982       1.7854     1.7788    1.00  0.31567
#> continentEurope:factor(year)1982    -6.9759     1.8336   -3.80  0.00015
#> continentOceania:factor(year)1982   -7.9409     5.7553   -1.38  0.16785
#> continentAmericas:factor(year)1987  -0.2267     1.9440   -0.12  0.90720
#> continentAsia:factor(year)1987       1.6858     1.7796    0.95  0.34363
#> continentEurope:factor(year)1987    -8.5955     1.8350   -4.68  3.0e-06
#> continentOceania:factor(year)1987   -9.4104     5.7554   -1.64  0.10223
#> continentAmericas:factor(year)1992   0.8213     1.9441    0.42  0.67276
#> continentAsia:factor(year)1992       2.6127     1.7803    1.47  0.14243
#> continentEurope:factor(year)1992    -7.7631     1.8346   -4.23  2.5e-05
#> continentOceania:factor(year)1992   -8.2567     5.7555   -1.43  0.15160
#> continentAmericas:factor(year)1997   2.1006     1.9443    1.08  0.28012
#> continentAsia:factor(year)1997       3.6483     1.7812    2.05  0.04070
#> continentEurope:factor(year)1997    -7.1773     1.8358   -3.91  9.6e-05
#> continentOceania:factor(year)1997   -7.5691     5.7557   -1.32  0.18867
#> continentAmericas:factor(year)2002   3.8379     1.9442    1.97  0.04854
#> continentAsia:factor(year)2002       5.0361     1.7814    2.83  0.00476
#> continentEurope:factor(year)2002    -6.1005     1.8369   -3.32  0.00092
#> continentOceania:factor(year)2002   -6.0287     5.7558   -1.05  0.29506
#> continentAmericas:factor(year)2007   3.2677     1.9444    1.68  0.09303
#> continentAsia:factor(year)2007       4.6483     1.7823    2.61  0.00919
#> continentEurope:factor(year)2007    -6.9223     1.8378   -3.77  0.00017
#> continentOceania:factor(year)2007   -6.4222     5.7558   -1.12  0.26468
#>
#> (Intercept)
```

```
#> log(gdpPercap)                     ***
#> log(pop)
#> continentAmericas                  ***
#> continentAsia                      ***
#> continentEurope                    ***
#> continentOceania                   ***
#> factor(year)1957
#> factor(year)1962                   **
#> factor(year)1967                   ***
#> factor(year)1972                   ***
#> factor(year)1977                   ***
#> factor(year)1982                   ***
#> factor(year)1987                   ***
#> factor(year)1992                   ***
#> factor(year)1997                   ***
#> factor(year)2002                   ***
#> factor(year)2007                   ***
#> continentAmericas:factor(year)1957
#> continentAsia:factor(year)1957
#> continentEurope:factor(year)1957
#> continentOceania:factor(year)1957
#> continentAmericas:factor(year)1962
#> continentAsia:factor(year)1962
#> continentEurope:factor(year)1962
#> continentOceania:factor(year)1962
#> continentAmericas:factor(year)1967
#> continentAsia:factor(year)1967
#> continentEurope:factor(year)1967
#> continentOceania:factor(year)1967
#> continentAmericas:factor(year)1972
#> continentAsia:factor(year)1972
#> continentEurope:factor(year)1972   *
#> continentOceania:factor(year)1972
#> continentAmericas:factor(year)1977
#> continentAsia:factor(year)1977
#> continentEurope:factor(year)1977   **
#> continentOceania:factor(year)1977
#> continentAmericas:factor(year)1982
#> continentAsia:factor(year)1982
#> continentEurope:factor(year)1982   ***
#> continentOceania:factor(year)1982
#> continentAmericas:factor(year)1987
#> continentAsia:factor(year)1987
#> continentEurope:factor(year)1987   ***
#> continentOceania:factor(year)1987
```

```
#> continentAmericas:factor(year)1992
#> continentAsia:factor(year)1992
#> continentEurope:factor(year)1992   ***
#> continentOceania:factor(year)1992
#> continentAmericas:factor(year)1997
#> continentAsia:factor(year)1997     *
#> continentEurope:factor(year)1997   ***
#> continentOceania:factor(year)1997
#> continentAmericas:factor(year)2002 *
#> continentAsia:factor(year)2002     **
#> continentEurope:factor(year)2002   ***
#> continentOceania:factor(year)2002
#> continentAmericas:factor(year)2007 .
#> continentAsia:factor(year)2007     **
#> continentEurope:factor(year)2007   ***
#> continentOceania:factor(year)2007
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 5.65 on 1642 degrees of freedom
#> Multiple R-squared:  0.816,  Adjusted R-squared:  0.809
#> F-statistic:  119 on 61 and 1642 DF,  p-value: <2e-16
```

Note that we wrapped the `year` variables into a `factor()` function. By default, R breaks up our variables into their different factor levels (as it will do whenever your regressors have factor levels).

If your data aren't factorized, you can tell `lm/glm` to factorize a variable (i.e. create dummy variables on the fly) by writing `factor()`

```
glm(formula = y ~ x1 + x2 + factor(x3), family = family(link = "link"),
            data = )
```

14.2.5 Formatting Regression Tables

Most papers report the results of regression analysis in some kind of table. Typically, this table includes the values of coefficients, standard errors, and significance levels from one or more models.

The `stargazer` package provides excellent tools to make and format regression tables automatically. It can also output summary statistics from a dataframe:

```
library(stargazer)
stargazer(gap, type = "text")
#>
#> ===========================================================================================
```

```
#> Statistic   N       Mean         St. Dev.      Min   Pctl(25)    Pctl(75)
#> -----------------------------------------------------------------------------------
#> year      1,704   1,980.000        17.300       1,952   1,966.0    1,993.0        2
#> pop       1,704 29,601,212.000 106,157,897.000 60,011  2,793,664 19,585,222.0 1,318
#> lifeExp   1,704     59.500         12.900      23.600   48.200     70.800       82
#> gdpPercap 1,704   7,215.000       9,857.000    241.000 1,202.000  9,325.000    113,
#> -----------------------------------------------------------------------------------
```

Let's say we want to report the results from three different models:

```
mod.1 <- lm(lifeExp ~ log(gdpPercap) + log(pop), data = gap)
mod.2 <- lm(lifeExp ~ log(gdpPercap) + log(pop) + continent, data = gap)
mod.3 <- lm(lifeExp ~ log(gdpPercap) + log(pop) + continent + year, data = gap)
```

`stargazer` can produce well-formatted tables that hold regression analysis results from all these models side-by-side.

```
stargazer(mod.1, mod.2, mod.3, title = "Regression Results", type = "text")
#> 
#> Regression Results
#> ===========================================================================================
#>                                                       Dependent variable:
#>                        --------------------------------------------------------------------
#>                                                             lifeExp
#>                                    (1)                        (2)
#> -------------------------------------------------------------------------------------------
#> log(gdpPercap)                   8.340***                   6.590***                     5
#>                                  (0.143)                    (0.182)
#> 
#> log(pop)                         1.280***                   0.866***
#>                                  (0.111)                    (0.111)
#> 
#> continentAmericas                                           6.170***                     8
#>                                                             (0.555)
#> 
#> continentAsia                                               4.670***                     6
#>                                                             (0.494)
#> 
#> continentEurope                                             8.560***                     1
#>                                                             (0.608)
#> 
#> continentOceania                                            8.350***                     1
#>                                                             (1.510)
#> 
#> year                                                                                     0
#> 
#> 
```

```
#> Constant                        -28.800***                  -12.000***                 -461.000***
#>                                  (2.080)                     (2.270)                    (17.000)
#>
#> ---------------------------------------------------------------------------------------------------
#> Observations                      1,704                       1,704                      1,704
#> R2                                0.677                       0.714                      0.798
#> Adjusted R2                       0.677                       0.713                      0.798
#> Residual Std. Error      7.340 (df = 1701)          6.920 (df = 1697)         5.810 (df = 1696
#> F Statistic         1,786.000*** (df = 2; 1701) 707.000*** (df = 6; 1697) 960.000*** (df = 7;
#> ===================================================================================================
#> Note:                                                                   *p<0.1; **p<0.05; ***p
```

Customization

`stargazer` is incredibly customizable. Let's say we wanted to:

- re-name our explanatory variables;
- remove information on the "Constant";

- only keep the number of observations from the summary statistics; and
- style the table to look like those in American Journal of Political Science.

```
stargazer(mod.1, mod.2, mod.3, title = "Regression Results", type = "text",
          covariate.labels  = c("GDP per capita, logged", "Population, logged", "Americas", "Asia
          omit = "Constant",
          keep.stat="n", style = "ajps")
#>
#> Regression Results
#> -----------------------------------------------------
#>                                   lifeExp
#>                          Model 1  Model 2   Model 3
#> -----------------------------------------------------
#> GDP per capita, logged 8.340*** 6.590*** 5.080***
#>                         (0.143)  (0.182)   (0.163)
#> Population, logged     1.280*** 0.866***   0.153
#>                         (0.111)  (0.111)   (0.097)
#> Americas                        6.170*** 8.740***
#>                                  (0.555)   (0.477)
#> Asia                            4.670*** 6.830***
#>                                  (0.494)   (0.423)
#> Europe                          8.560*** 12.300***
#>                                  (0.608)   (0.529)
#> Oceania                         8.350*** 12.500***
#>                                  (1.510)   (1.280)
#> Year                                      0.238***
```

```
#>                                            (0.009)
#> N                        1704     1704     1704
#> --------------------------------------------------
#> ***p < .01; **p < .05; *p < .1
```

Check out `?stargazer` to see more options.

Output Types

Once we like the look of our table, we can output/export it in a number of ways. The `type` argument specifies what output the command should produce. Possible values are:

- `"latex"` for LaTeX code,
- `"html"` for HTML code,
- `"text"` for ASCII text output (what we used above).

Let's say we're using LaTeX to typeset our paper. We can output our regression table in LaTeX:

```
stargazer(mod.1, mod.2, mod.3, title = "Regression Results", type = "latex",
          covariate.labels  = c("GDP per capita, logged", "Population, logged", "Ameri
          omit = "Constant",
          keep.stat="n", style = "ajps")
#> 
#> % Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: h
#> % Date and time: Wed, Nov 13, 2019 - 11:18:29
#> \begin{table}[!htbp] \centering
#>   \caption{Regression Results}
#>   \label{}
#> \begin{tabular}{@{\extracolsep{5pt}}lccc}
#> \\[-1.8ex]\hline \\[-1.8ex]
#> \\[-1.8ex] & \multicolumn{3}{c}{\textbf{lifeExp}} \\
#> \\[-1.8ex] & \textbf{Model 1} & \textbf{Model 2} & \textbf{Model 3}\\
#> \hline \\[-1.8ex]
#>  GDP per capita, logged & 8.340$^{***}$ & 6.590$^{***}$ & 5.080$^{***}$ \\
#>   & (0.143) & (0.182) & (0.163) \\
#>   Population, logged & 1.280$^{***}$ & 0.866$^{***}$ & 0.153 \\
#>   & (0.111) & (0.111) & (0.097) \\
#>   Americas &  & 6.170$^{***}$ & 8.740$^{***}$ \\
#>   &  & (0.555) & (0.477) \\
#>   Asia &  & 4.670$^{***}$ & 6.830$^{***}$ \\
#>   &  & (0.494) & (0.423) \\
#>   Europe &  & 8.560$^{***}$ & 12.300$^{***}$ \\
#>   &  & (0.608) & (0.529) \\
#>   Oceania &  & 8.350$^{***}$ & 12.500$^{***}$ \\
```

```
#>   &  & (1.510) & (1.280) \\
#>   Year &  &  & 0.238$^{***}$ \\
#>   &  &  & (0.009) \\
#>  N & 1704 & 1704 & 1704 \\
#> \hline \\[-1.8ex]
#> \multicolumn{4}{l}{$^{***}$p $<$ .01; $^{**}$p $<$ .05; $^{*}$p $<$ .1} \\
#> \end{tabular}
#> \end{table}
```

To include the produced tables in our paper, we can simply insert this stargazer LaTeX output into the publication's TeX source.

Alternatively, you can use the `out` argument to save the output in a .tex or .txt file:

```
stargazer(mod.1, mod.2, mod.3, title = "Regression Results", type = "latex",
          covariate.labels  = c("GDP per capita, logged", "Population, logged", "Americas", "Asia
          omit = "Constant",
          keep.stat="n", style = "ajps",
          out = "regression-table.txt")
```

To include stargazer tables in Microsoft Word documents (e.g., .doc or .docx), use the following procedure:

- Use the `out` argument to save output into an `.html` file.
- Open the resulting file in your web browser.
- Copy and paste the table from the web browser to your Microsoft Word document.

```
stargazer(mod.1, mod.2, mod.3, title = "Regression Results", type = "html",
          covariate.labels  = c("GDP per capita, logged", "Population, logged", "Americas", "Asia
          omit = "Constant",
          keep.stat="n", style = "ajps",
          out = "regression-table.html")
```

14.2.6 Challenges

Challenge 1.

Fit two linear regression models from the gapminder data, where the outcome is `lifeExp` and the explanatory variables are `log(pop)`, `log(gdpPercap)`, and `year`. In one model, treat `year` as a numeric variable. In the other, factorize the `year` variable. How do you interpret each model?

Challenge 2.

Fit a logistic regression model where the outcome is whether `lifeExp` is greater than or less than 60 years, exploring the use of different predictors.

Challenge 3.

Using `stargazer`, format a table reporting the results from the three models you created above (two linear regressions and one logistic).

Chapter 15

Strings and Regular Expressions

This unit focuses on chracter (or "string") data. We'll explore:

1. **string basics**, like concatinating and subsettings.
2. **regular expressions**, a powerful cross-language tool for working with string data.
3. **common tools**, that take regex and apply them to real problems.

This chapter will focus on the **`stringr`** package for string manipulation. `stringr` is not part of the core `tidyverse` because you don't always have textual data, so we need to load it explicitly.

```
library(tidyverse)
library(stringr)
```

15.1 String Basics

15.1.1 Creating Strings

You can create strings with either single quotes or double quotes. Unlike other languages, there is no difference in behavior. I recommend always using `"`, unless you want to create a string that contains multiple `"`.

```
string1 <- "This is a string"
string2 <- 'If I want to include a "quote" inside a string, I use single quotes'
```

15.1.2 Escape and Special Characters

Single and double quotes are known as "metacharacters," meaning that they have special meaning to the R language. To include a literal single or double quote in a string you can use \ to "escape" it:

```
double_quote <- "\"" # or '"'
single_quote <- '\'' # or "'"
```

That means if you want to include a literal backslash, you'll need to double it up: `"\\"`.

Beware that the printed representation of a string is not the same as string itself, because the printed representation shows the escapes. To see the raw contents of the string, use `writeLines()`:

```
x <- c("\"", "\\")
x
#> [1] "\"" "\\"
writeLines(x)
#> "
#> \
```

There are a handful of other special characters. The most common are `"\n"`, newline, and `"\t"`, tab, but you can see the complete list by requesting help on ": `?'"'`, or `?"'"`. You'll also sometimes see strings like `"\u00b5"`, this is a way of writing non-English characters that works on all platforms:

```
x <- "\u00b5"
x
#> [1] "µ"
```

Multiple strings are often stored in a character vector, which you can create with `c()`:

```
c("one", "two", "three")
#> [1] "one"   "two"   "three"
```

15.1.3 String length

Base R contains many functions to work with strings but we'll avoid them because they can be inconsistent, which makes them hard to remember. Instead we'll use functions from `stringr`. These have more intuitive names, and all start with `str_`. For example, `str_length()` tells you the number of characters in a string:

```
str_length(c("a", "R for data science", NA))
#> [1]  1 18 NA
```

The common `str_` prefix is particularly useful if you use RStudio, because typing `str_` will trigger autocomplete, allowing you to see all `stringr` functions:

15.1.4 Combining strings

To combine two or more strings, use `str_c()`:

```
str_c("x", "y")
#> [1] "xy"
str_c("x", "y", "z")
#> [1] "xyz"
```

Use the `sep` argument to control how they're separated:

```
str_c("x", "y", sep = ", ")
#> [1] "x, y"
```

`str_c()` is vectorised, and it automatically recycles shorter vectors to the same length as the longest:

```
x <- c("a", "b", "c")
str_c("prefix-", x)
#> [1] "prefix-a" "prefix-b" "prefix-c"
```

To collapse a vector of strings into a single string, use `collapse`:

```
x <- c("x", "y", "z")
str_c(x, collapse = ", ")
#> [1] "x, y, z"
```

15.1.5 Subsetting strings

You can extract parts of a string using `str_sub()`. As well as the string, `str_sub()` takes `start` and `end` arguments, which give the (inclusive) position of the substring:

```
x <- c("Rochelle is the Greatest")
str_sub(x, 1, 8)
#> [1] "Rochelle"

# negative numbers count backwards from end
str_sub(x, -8, -1)
#> [1] "Greatest"
```

Note that `str_sub()` won't fail if the string is too short: it will just return as much as possible:

```
str_sub("a", 1, 3)
#> [1] "a"
```

You can also use the assignment form of `str_sub()` to modify strings:

```
x <- c("Rochelle is the Greatest")
str_sub(x, 1, 1) <- str_to_lower(str_sub(x, 1, 1))
x
#> [1] "rochelle is the Greatest"
```

15.1.6 Locales

Above I used `str_to_lower()` to change the text to lower case. You can also use `str_to_upper()` or `str_to_title()`. However, changing case is more complicated than it might at first appear because different languages have different rules for changing case. You can pick which set of rules to use by specifying a locale:

```
# Turkish has two i's: with and without a dot, and it
# has a different rule for capitalising them:
str_to_upper(c("i", "ı"))
#> [1] "I" "I"
str_to_upper(c("i", "ı"), locale = "tr")
#> [1] "İ" "I"
```

The locale is specified as a ISO 639 language code, which is a two or three letter abbreviation. If you don't already know the code for your language, Wikipedia has a good list. If you leave the locale blank, it will use the current locale, as provided by your operating system.

Another important operation that's affected by the locale is sorting. The base R `order()` and `sort()` functions sort strings using the current locale. If you want robust behaviour across different computers, you may want to use `str_sort()` and `str_order()` which take an additional `locale` argument:

```
x <- c("apple", "eggplant", "banana")
str_sort(x, locale = "en")  # English
#> [1] "apple"    "banana"   "eggplant"
str_sort(x, locale = "haw") # Hawaiian
#> [1] "apple"    "eggplant" "banana"
```

Challenges

1. In code that doesn't use `stringr`, you'll often see `paste()` and `paste0()`. What's the difference between the two functions? What `stringr` function

are they equivalent to? How do the functions differ in their handling of `NA`?

2. In your own words, describe the difference between the `sep` and `collapse` arguments to `str_c()`.

3. Use `str_length()` and `str_sub()` to extract the middle character from a string. What will you do if the string has an even number of characters?

4. What does `str_trim()` do? What's the opposite of `str_trim()`?

15.2 Regular expressions

Regular expressions are a very terse language that allow you to describe patterns in strings. They take a little while to get your head around, but once you understand them, you'll find them extremely useful.

To learn regular expressions, we'll use `str_view()` and `str_view_all()`. These functions take a character vector and a regular expression, and show you how they match. We'll start with very simple regular expressions and then gradually get more and more complicated. Once you've mastered pattern matching, you'll learn how to apply those ideas with various stringr functions.

15.2.1 Basic matches

The simplest patterns match exact strings:

```
x <- c("apple", "banana", "pear")
str_view(x, "an")
#> PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed,
```

The next step up in complexity is `.`, which matches any character (except a newline):

```
x <- c("apple", "banana", "pear")
str_view(x, ".a.")
```

15.2.2 Escape Characters

If "`.`" matches any character, how do you match the character "`.`"? You need to use an "escape" to tell the regular expression you want to match it exactly, not use its special behaviour.

Regexps use the backslash, `\`, to escape special behaviour. So to match an `.`, you need the regexp `\.`. Unfortunately this creates a problem. We use *strings* to represent regular expressions, and \ is also used as an escape symbol in strings. So to create the regular expression `\.` we need the string `"\\."`.

```
# To create the regular expression, we need \\
dot <- "\\."
# But the expression itself only contains one:
writeLines(dot)
#> \.
```

In this lesson, I'll write regular expression as `\.` and strings that represent the regular expression as `"\\."`.

15.2.3 Anchors

By default, regular expressions will match any part of a string. It's often useful to *anchor* the regular expression so that it matches from the start or end of the string. You can use:

- `^` to match the start of the string.
- `$` to match the end of the string.

```
x <- c("apple", "banana", "pear")
str_view(x, "^a")
```

```
str_view(x, "a$")
```

To remember which is which, try this mnemonic which I learned from Evan Misshula: if you begin with power (`^`), you end up with money (`$`).

To force a regular expression to only match a complete string, anchor it with both `^` and `$`:

```
x <- c("apple pie", "apple", "apple cake")
str_view(x, "apple")
```

```
str_view(x, "^apple$")
```

15.2.4 Character classes and alternatives

There are a number of special patterns that match more than one character. You've already seen `.`, which matches any character apart from a newline. There are four other useful tools:

- `\d`: matches any digit.
- `\s`: matches any whitespace (e.g. space, tab, newline).
- `[abc]`: matches a, b, or c.
- `[^abc]`: matches anything except a, b, or c.

Remember, to create a regular expression containing `\d` or `\s`, you'll need to escape the `\` for the string, so you'll type `"\\d"` or `"\\s"`.

A character class containing a single character is a nice alternative to backslash escapes when you want to include a single metacharacter in a regex. Many people find this more readable.

```
# Look for a literal character that normally has special meaning in a regex
x <- c("abc", "a.c", "a*c", "a c")
str_view(x, "a[.]c")
```

```
str_view(x, ".[*]c")
```

```
str_view(x, "a[ ]")
```

This works for most (but not all) regex metacharacters: `$ . | ? * + ( ) [ {`. Unfortunately, a few characters have special meaning even inside a character class and must be handled with backslash escapes: `] \ ^` and `-`.

You can use *alternation* to pick between one or more alternative patterns. For example, `abc|deaf` will match either '"abc"', or `"deaf"`.

Like with mathematical expressions, if precedence ever gets confusing, use parentheses to make it clear what you want:

```
x <- c("grey", "gray")
str_view(x, "gr(e|a)y")
```

Challenges

Create regular expressions to find all words that:

1. Start with a vowel.
2. That only contain consonants. (Hint: thinking about matching "not"-vowels.)
3. End with `ed`, but not with `eed`.
4. End with `ing` or `ise`.

15.2.5 Repetition

The next step up in power involves controlling how many times a pattern matches:

- `?`: 0 or 1
- `+`: 1 or more
- `*`: 0 or more

```
x <- "1888 is the longest year in Roman numerals: MDCCCLXXXVIII"
str_view(x, "CC?")
```

```
str_view(x, "CC+")
```

```
str_view(x, 'C[LX]+')
```

15.2.6 Regex Resources

For more information on regular expressions, see:

1. this tutorial
2. this cheatsheet

15.3 Common Tools

Now that you've learned the basics of regular expressions, it's time to learn how to apply them to real problems. In this section you'll learn a wide array of `stringr` functions that let you:

- Determine which strings match a pattern.
- Find the positions of matches.
- Extract the content of matches.
- Replace matches with new values.
- Split a string based on a match.

15.3.1 Detect matches

To determine if a character vector matches a pattern, use `str_detect()`. It returns a logical vector the same length as the input:

```
x <- c("apple", "banana", "pear")
str_detect(x, "e")
#> [1]  TRUE FALSE  TRUE
```

Remember that when you use a logical vector in a numeric context, `FALSE` becomes 0 and `TRUE` becomes 1. That makes `sum()` and `mean()` useful if you want to answer questions about matches across a larger vector:

```
# see common words
words[1:10]
#>  [1] "a"        "able"     "about"    "absolute" "accept"   "account"
#>  [7] "achieve"  "across"   "act"      "active"
# How many common words start with t?
sum(str_detect(words, "^t"))
```

```
#> [1] 65
# What proportion of common words end with a vowel?
mean(str_detect(words, "[aeiou]$"))
#> [1] 0.277
```

A common use of `str_detect()` is to select the elements that match a pattern. You can do this with logical subsetting, or the convenient `str_subset()` wrapper:

```
words[str_detect(words, "x$")]
#> [1] "box" "sex" "six" "tax"
str_subset(words, "x$")
#> [1] "box" "sex" "six" "tax"
```

Typically, however, your strings will be one column of a data frame, and you'll want to use filter instead:

```
df <- data.frame(
  i = seq_along(words),
  word = words
)
df %>%
  filter(str_detect(word, "x$"))
#>     i word
#> 1 108  box
#> 2 747  sex
#> 3 772  six
#> 4 841  tax
```

A variation on `str_detect()` is `str_count()`: rather than a simple yes or no, it tells you how many matches there are in a string:

```
x <- c("apple", "banana", "pear")
str_count(x, "a")
#> [1] 1 3 1
# On average, how many vowels per word?
mean(str_count(words, "[aeiou]"))
#> [1] 1.99
```

It's natural to use `str_count()` with `mutate()`:

```
df1 <- df %>%
  mutate(
    vowels = str_count(word, "[aeiou]"),
    consonants = str_count(word, "[^aeiou]")
  )

head(df1)
```

```
#>   i     word vowels consonants
#> 1 1        a      1          0
#> 2 2     able      2          2
#> 3 3    about      3          2
#> 4 4 absolute      4          4
#> 5 5   accept      2          4
#> 6 6  account      3          4
```

Challenges

For each of the following challenges, try solving it by using both a single regular expression, and a combination of multiple `str_detect()` calls.

1. Find all words that start or end with `x`.
2. Find all words that start with a vowel and end with a consonant.

15.3.2 Extract matches

To extract the actual text of a match, use `str_extract()`. To show that off, we're going to need a more complicated example. I'm going to use the Harvard sentences. These are provided in `stringr::sentences`:

```
length(sentences)
#> [1] 720
head(sentences)
#> [1] "The birch canoe slid on the smooth planks."
#> [2] "Glue the sheet to the dark blue background."
#> [3] "It's easy to tell the depth of a well."
#> [4] "These days a chicken leg is a rare dish."
#> [5] "Rice is often served in round bowls."
#> [6] "The juice of lemons makes fine punch."
```

Imagine we want to find all sentences that contain a color. We first create a vector of color names, and then turn it into a single regular expression:

```
colors <- c("red", "orange", "yellow", "green", "blue", "purple")
color_match <- str_c(colors, collapse = "|")
color_match
#> [1] "red|orange|yellow|green|blue|purple"
```

Now we can select the sentences that contain a color, and then extract the color to figure out which one it is:

```
# find sentences with colors
has_color <- str_subset(sentences, color_match)
```

```
head(has_color)
#> [1] "Glue the sheet to the dark blue background."
#> [2] "Two blue fish swam in the tank."
#> [3] "The colt reared and threw the tall rider."
#> [4] "The wide road shimmered in the hot sun."
#> [5] "See the cat glaring at the scared mouse."
#> [6] "A wisp of cloud hung in the blue air."

# extract the color
matches <- str_extract(has_color, color_match)
head(matches)
#> [1] "blue" "blue" "red"  "red"  "red"  "blue"
```

Note that `str_extract()` only extracts the first match. This is a common pattern for stringr functions, because working with a single match allows you to use much simpler data structures. To get all matches, use `str_extract_all()`. It returns a list:

```
all_colors <- str_extract_all(has_color, color_match)
all_colors[15:20]
#> [[1]]
#> [1] "red"
#>
#> [[2]]
#> [1] "red"
#>
#> [[3]]
#> [1] "red"
#>
#> [[4]]
#> [1] "blue"
#>
#> [[5]]
#> [1] "red"
#>
#> [[6]]
#> [1] "blue" "red"
```

If you use `simplify = TRUE`, `str_extract_all()` will return a matrix with short matches expanded to the same length as the longest:

```
str_extract_all(has_color, color_match, simplify = TRUE)
#>       [,1]     [,2]
#>  [1,] "blue"   ""
#>  [2,] "blue"   ""
#>  [3,] "red"    ""
#>  [4,] "red"    ""
```

```
#>  [5,] "red"    ""
#>  [6,] "blue"   ""
#>  [7,] "yellow" ""
#>  [8,] "red"    ""
#>  [9,] "red"    ""
#> [10,] "green"  ""
#> [11,] "red"    ""
#> [12,] "red"    ""
#> [13,] "blue"   ""
#> [14,] "red"    ""
#> [15,] "red"    ""
#> [16,] "red"    ""
#> [17,] "red"    ""
#> [18,] "blue"   ""
#> [19,] "red"    ""
#> [20,] "blue"   "red"
#> [21,] "red"    ""
#> [22,] "green"  ""
#> [23,] "red"    ""
#> [24,] "red"    ""
#> [25,] "red"    ""
#> [26,] "red"    ""
#> [27,] "red"    ""
#> [28,] "red"    ""
#> [29,] "green"  ""
#> [30,] "red"    ""
#> [31,] "green"  ""
#> [32,] "red"    ""
#> [33,] "purple" ""
#> [34,] "green"  ""
#> [35,] "red"    ""
#> [36,] "red"    ""
#> [37,] "red"    ""
#> [38,] "red"    ""
#> [39,] "red"    ""
#> [40,] "blue"   ""
#> [41,] "red"    ""
#> [42,] "blue"   ""
#> [43,] "red"    ""
#> [44,] "red"    ""
#> [45,] "red"    ""
#> [46,] "red"    ""
#> [47,] "green"  ""
#> [48,] "green"  ""
#> [49,] "green"  "red"
```

```
#> [50,] "red"    ""
#> [51,] "red"    ""
#> [52,] "yellow" ""
#> [53,] "red"    ""
#> [54,] "orange" "red"
#> [55,] "red"    ""
#> [56,] "red"    ""
#> [57,] "red"    ""
```

Challenges

In the previous example, you might have noticed that the regular expression matched "flickered", which is not a color. Modify the regex to fix the problem.

15.3.3 Replacing matches

`str_replace()` and `str_replace_all()` allow you to replace matches with new strings. The simplest use is to replace a pattern with a fixed string:

```
x <- c("apple", "pear", "banana")
str_replace(x, "[aeiou]", "-") # replace the first instance of a match
#> [1] "-pple"  "p-ar"   "b-nana"
str_replace_all(x, "[aeiou]", "-") # replace all instances of a match
#> [1] "-ppl-"  "p--r"   "b-n-n-"
```

With `str_replace_all()` you can perform multiple replacements by supplying a named vector:

```
x <- c("1 house", "2 cars", "3 people")
str_replace_all(x, c("1" = "one", "2" = "two", "3" = "three"))
#> [1] "one house"    "two cars"     "three people"
```

15.3.4 Splitting

Use `str_split()` to split a string up into pieces. For example, we could split sentences into words:

```
sentences %>%
  head(5) %>%
  str_split(" ")
#> [[1]]
#> [1] "The"     "birch"   "canoe"   "slid"    "on"      "the"     "smooth"
#> [8] "planks."
#>
```

```
#> [[2]]
#> [1] "Glue"        "the"         "sheet"       "to"          "the"
#> [6] "dark"        "blue"        "background."
#>
#> [[3]]
#> [1] "It's"   "easy"   "to"     "tell"   "the"    "depth" "of"     "a"      "well."
#>
#> [[4]]
#> [1] "These"   "days"    "a"       "chicken" "leg"     "is"      "a"
#> [8] "rare"    "dish."
#>
#> [[5]]
#> [1] "Rice"   "is"     "often"  "served" "in"     "round"  "bowls."
```

Like the other stringr functions that return a list, you can use `simplify = TRUE` to return a matrix:

```
sentences %>%
  head(5) %>%
  str_split(" ", simplify = TRUE)
#>      [,1]    [,2]    [,3]    [,4]      [,5]  [,6]    [,7]
#> [1,] "The"   "birch" "canoe" "slid"    "on"  "the"   "smooth"
#> [2,] "Glue"  "the"   "sheet" "to"      "the" "dark"  "blue"
#> [3,] "It's"  "easy"  "to"    "tell"    "the" "depth" "of"
#> [4,] "These" "days"  "a"     "chicken" "leg" "is"    "a"
#> [5,] "Rice"  "is"    "often" "served"  "in"  "round" "bowls."
#>      [,8]          [,9]
#> [1,] "planks."     ""
#> [2,] "background." ""
#> [3,] "a"           "well."
#> [4,] "rare"        "dish."
#> [5,] ""            ""
```

You can also request a maximum number of pieces:

```
fields <- c("Name: Hadley", "Country: NZ", "Age: 35")
fields %>% str_split(": ", n = 2, simplify = TRUE)
#>      [,1]      [,2]
#> [1,] "Name"    "Hadley"
#> [2,] "Country" "NZ"
#> [3,] "Age"     "35"
```

Instead of splitting up strings by patterns, you can also split up by character, line, sentence and word `boundary()`s:

```
x <- "This is a sentence.  This is another sentence."
str_view_all(x, boundary("word"))
```

```r
str_split(x, boundary("word"))[[1]]
#> [1] "This"     "is"       "a"        "sentence" "This"     "is"
#> [7] "another"  "sentence"
```

Challenges

1. Split up a string like `"apples, pears, and bananas"` into individual components.
2. What does splitting with an empty string (`""`) do? Experiment, and then read the documentation.

15.4 Other types of patterns

When you use a pattern that's a string, it's automatically wrapped into a call to `regex()`:

```r
# The regular call:
str_view(fruit, "nana")
# Is shorthand for
str_view(fruit, regex("nana"))
```

You can use the other arguments of `regex()` to control details of the match:

- `ignore_case = TRUE` allows characters to match either their uppercase or lowercase forms. This always uses the current locale.

  ```r
  bananas <- c("banana", "Banana", "BANANA")
  str_view(bananas, "banana")
  ```

  ```r
  str_view(bananas, regex("banana", ignore_case = TRUE))
  ```

- `multiline = TRUE` allows `^` and `$` to match the start and end of each line rather than the start and end of the complete string.

  ```r
  x <- "Line 1\nLine 2\nLine 3"
  str_extract_all(x, "^Line")[[1]]
  #> [1] "Line"
  str_extract_all(x, regex("^Line", multiline = TRUE))[[1]]
  #> [1] "Line" "Line" "Line"
  ```

15.4.1 stringi

stringr is built on top of the **stringi** package. stringr is useful when you're learning because it exposes a minimal set of functions, which have been carefully picked to handle the most common string manipulation functions. stringi, on the other hand, is designed to be comprehensive. It contains almost every function you might ever need.

If you find yourself struggling to do something in stringr, it's worth taking a look at stringi. The packages work very similarly, so you should be able to translate your stringr knowledge in a natural way. The main difference is the prefix: `str_` vs. `stri_`.

Challenges

Find the stringi functions that:

1. Count the number of words.
2. Find duplicated strings.
3. Generate random text.

Acknowledgments

This page was adapted from the following sources:

1. R for Data Science licensed under Creative Commons Attribution-NonCommercial-NoDerivs 3.0

Chapter 16

Programming in R

This unit covers some more advanced programming in R - namely:

1. Conditional Flow
2. Functions
3. Iteration

Mastering these skills will make you virtually invincible in R!

Note that these concepts are **not specific to R**. While the syntax might vary, the basic idea of flow, functions, and iteration are common across all scripting languages. So if you ever think of picking up Python or something else, it's critical to familiarize yourself with these concepts.

16.1 Conditional Flow

Sometimes you only want to execute code if a certain condition is met. To do that, we use an **if-else statement.** It looks like this:

```
if (condition) {
  # code executed when condition is TRUE
} else {
  # code executed when condition is FALSE
}
```

`condition` is a statement that must always evaluate to either `TRUE` or `FALSE`. This is similar to `filter()`, except `condition` can only be a single value (i.e. a vector of length 1), whereas `filter()` works for entire vectors (or columns).

Let's look at a simple example:

```
age = 84
if (age > 60) {
    print("OK Boomer")
} else {
    print("But you don't look like a professor!")
}
#> [1] "OK Boomer"
```

We refer to the first `print` command as the first *branch*.

Let's change the `age` variable to execute the second branch:

```
age = 20
if (age > 60) {
    print("OK Boomer")
} else {
    print("But you don't look like a professor!")
}
#> [1] "But you don't look like a professor!"
```

16.1.1 Multiple Conditions

You can chain conditional statements together:

```
if (this) {
  # do that
} else if (that) {
  # do something else
} else {
  # do something completely different
}
```

16.1.2 Complex Statements

We can generate more complex conditional statements with boolean operators like `&` and `|`:

```
age = 45

if (age > 60) {
    print("OK Boomer")
} else if (age < 60 & age > 40) {
    print("How's the midlife crisis?")
} else {
    print("But you don't look like a professor!")
```

```
}
#> [1] "How's the midlife crisis?"
```

16.1.3 Code Style

Both `if` and `function` should (almost) always be followed by squiggly brackets (`{}`), and the contents should be indented. This makes it easier to see the hierarchy in your code by skimming the left-hand margin.

An opening curly brace should never go on its own line and should always be followed by a new line. A closing curly brace should always go on its own line, unless it's followed by else. Always indent the code inside curly braces.

```
# Bad
if (y < 0 && debug)
message("Y is negative")

if (y == 0) {
  log(x)
}
else {
  y ^ x
}

# Good
if (y < 0 && debug) {
  message("Y is negative")
}

if (y == 0) {
  log(x)
} else {
  y ^ x
}
```

16.1.4 `if` vs. `if_else`

Because if-else conditional statements like the ones outlined above must always resolve to a single `TRUE` or `FALSE`, they cannot be used for vector operations. Vector operations are where you make multiple comparisons simultaneously for each value stored inside a vector.

Consider the `gapminder` data and imagine you wanted to create a new column identifying whether or not a country-year observation has a life expectancy of at least 35.

```
gap <- read.csv("Data/gapminder-FiveYearData.csv")
head(gap)
#>       country year      pop continent lifeExp gdpPercap
#> 1 Afghanistan 1952  8425333      Asia    28.8       779
#> 2 Afghanistan 1957  9240934      Asia    30.3       821
#> 3 Afghanistan 1962 10267083      Asia    32.0       853
#> 4 Afghanistan 1967 11537966      Asia    34.0       836
#> 5 Afghanistan 1972 13079460      Asia    36.1       740
#> 6 Afghanistan 1977 14880372      Asia    38.4       786
```

This sounds like a classic if-else operation. For each observation, if `lifeExp` is greater than or equal to `35`, then the value in the new column should be `1`. Otherwise, it should be `0`. But what happens if we try to implement this using an if-else operation like above?

```
gap_if <- gap %>%
   mutate(life.35 = if(lifeExp >= 35){
     1
   } else {
     0
   })
#> Warning in if (lifeExp >= 35) {: the condition has length > 1 and only the
#> first element will be used

head(gap_if)
#>       country year      pop continent lifeExp gdpPercap life.35
#> 1 Afghanistan 1952  8425333      Asia    28.8       779       0
#> 2 Afghanistan 1957  9240934      Asia    30.3       821       0
#> 3 Afghanistan 1962 10267083      Asia    32.0       853       0
#> 4 Afghanistan 1967 11537966      Asia    34.0       836       0
#> 5 Afghanistan 1972 13079460      Asia    36.1       740       0
#> 6 Afghanistan 1977 14880372      Asia    38.4       786       0
```

This did not work correctly. Because `if()` can only handle a single `TRUE/FALSE` value, it only checked the first row of the data frame. That row contained `28.801`, so it generated a vector of length 1704 with each value being `0`.

Because we in fact want to make this if-else comparison 1704 times, we should instead use **`if_else()`**. This **vectorizes** the if-else comparison and makes a separate comparison for each row of the data frame. This allows us to correctly generate this new column.

```
gap_ifelse <- gap %>%
  mutate(life.35 = if_else(lifeExp >= 35, 1, 0))

head(gap_ifelse)
#>       country year      pop continent lifeExp gdpPercap life.35
```

```
#> 1 Afghanistan 1952  8425333      Asia    28.8       779        0
#> 2 Afghanistan 1957  9240934      Asia    30.3       821        0
#> 3 Afghanistan 1962 10267083      Asia    32.0       853        0
#> 4 Afghanistan 1967 11537966      Asia    34.0       836        0
#> 5 Afghanistan 1972 13079460      Asia    36.1       740        1
#> 6 Afghanistan 1977 14880372      Asia    38.4       786        1
```

16.2 Functions

Functions are the basic building blocks of programs. Think of them as "mini-scripts" or "tiny commands." We've already used dozens of functions created by others (e.g. `filter()`, `mean()`.)

This lesson teaches you how to write you own functions, and why you would want to do so. The details are pretty simple, but this is one of those ideas where it's good to get lots of practice!

16.2.1 Why Write Functions?

Functions allow you to automate common tasks in a more powerful and general way than copy-and-pasting. For example, take a look at the following code:

```
df <- data.frame(
  a = rnorm(10),
  b = rnorm(10),
  c = rnorm(10),
  d = rnorm(10)
)

df$a <- (df$a - min(df$a)) / (max(df$a) - min(df$a))
df$b <- (df$b - min(df$b)) / (max(df$b) - min(df$a))
df$c <- (df$c - min(df$c)) / (max(df$c) - min(df$c))
df$d <- (df$d - min(df$d)) / (max(df$d) - min(df$d))
```

You might be able to puzzle out that this rescales each column to have a range from 0 to 1. But did you spot the mistake? I made an error when copying-and-pasting the code for `df$b`: I forgot to change an `a` to a `b`.

Functions have a number of advantages over this "copy-and-paste" approach:

- **They are easy to reuse.** If you need to change things, you only have to update code in one place, instead of many.
- **They are self-documenting.** Functions name pieces of code the way variables name strings and numbers. Give your function a good name and

you will easily remember the function and its purpose.

- **They are easier to debug.** There are fewer chances to make mistakes because the code only exists in one location (i.e. updating a variable name in one place, but not in another).

16.2.2 Anatomy of a Function

Functions have three key components:

1. A **name**. This should be informative and describe what the function does.
2. The **arguments**, or list of inputs, to the function. They go inside the parentheses in `function()`.
3. The **body**. This is the block of code within `{}` that immediately follows `function(...)`, and is the code that you develop to perform the action described in the name using the arguments you provide.

```
my_function <- function(x, y){
  # do
  # something
  # here
  return(result)
}
```

In this example, `my_function` is the **name** of the function, `x` and `y` are the **arguments**, and the stuff inside the `{}` is the **body**.

16.2.3 Writing a Function

Let's re-write the scaling code above as a function. To write a function you need to first analyze the code. How many inputs does it have?

```
df <- data.frame(
  a = rnorm(10),
  b = rnorm(10),
  c = rnorm(10),
  d = rnorm(10)
)

(df$a - min(df$a)) / (max(df$a) - min(df$a))
#>  [1] 0.289 0.751 0.000 0.678 0.853 1.000 0.172 0.611 0.612 0.601
```

This code only has one input: `df$a`. To make the inputs more clear, it's a good idea to rewrite the code using temporary variables with general names. Here this code only requires a single numeric vector, which I'll call `x`:

```
x <- df$a
(x - min(x)) / (max(x) - min(x))
#>  [1] 0.289 0.751 0.000 0.678 0.853 1.000 0.172 0.611 0.612 0.601
```

There is some duplication in this code. We're computing the range of the data three times, so it makes sense to do it in one step:

```
rng <- range(x)
rng
#> [1] -2.44  1.15

(x - rng[1]) / (rng[2] - rng[1])
#>  [1] 0.289 0.751 0.000 0.678 0.853 1.000 0.172 0.611 0.612 0.601
```

Pulling out intermediate calculations into named variables is a good practice because it becomes more clear what the code is doing. Now that I've simplified the code, and checked that it still works, I can turn it into a function:

```
rescale01 <- function(x) {
  rng <- range(x)
  scaled <- (x - rng[1]) / (rng[2] - rng[1])
  return(scaled)
}
```

Note the overall process: I only made the function after I'd figured out how to make it work with a simple input. It's easier to start with working code and turn it into a function; it's harder to create a function and then try to make it work.

At this point it's a good idea to check your function with a few different inputs:

```
rescale01(c(-10, 0, 10))
#> [1] 0.0 0.5 1.0

rescale01(c(1, 2, 3, 5))
#> [1] 0.00 0.25 0.50 1.00
```

16.2.4 Using a Function

Two important points about using (or *calling**) functions:

1. Notice that when we **call** a function, we're passing a value into it that is assigned to the parameter we defined when writing the function. In this case, the parameter `x` is automatically assigned to `c(-10, 0, 10)`.

2. When using functions, by default the returned object is merely printed to the screen. If you want it saved, you need to assign it to an object.

Let's see if we can simplify the original example with our brand new function:

```
df$a <- rescale01(df$a)
df$b <- rescale01(df$b)
df$c <- rescale01(df$c)
df$d <- rescale01(df$d)
```

Compared to the original, this code is easier to understand and we've eliminated one class of copy-and-paste errors. There is still quite a bit of duplication since we're doing the same thing to multiple columns. We'll learn how to eliminate that duplication in the lesson on iteration.

Another advantage of functions is that if our requirements change, we only need to make the change in one place. For example, we might discover that some of our variables include `NA` values, and `rescale01()` fails:

```
rescale01(c(1, 2, NA, 3, 4, 5))
#> [1] NA NA NA NA NA NA
```

Because we've extracted the code into a function, we only need to make the fix in one place:

```
rescale01 <- function(x) {
  rng <- range(x, na.rm = TRUE)
  (x - rng[1]) / (rng[2] - rng[1])
}
rescale01(c(1, 2, NA, 3, 4, 5))
#> [1] 0.00 0.25   NA 0.50 0.75 1.00
```

16.2.5 Variable Scope

Analyze the following function:

1. Identify the name, arguments, and body
2. What does it do?
3. If `a = 3` and `b = 4`, what should we expect the output to be?

```
pythagorean <- function(a, b){
  hypotenuse <- sqrt(a^2 + b^2)
  return(hypotenuse)
}
```

Now take a look at the following code:

```
pythagorean(a = 3, b = 4)
#> [1] 5

hypotenuse
#> Error in eval(expr, envir, enclos): object 'hypotenuse' not found
```

Why does this generate an error? Why can we not see the results of `hypotenuse`? After all, it was generated by `pythagorean`, right?

When you call a function, a temporary workspace is set up that will be destroyed when the function returns, either by:

1. getting to the end, or
2. an explicit return statement

So think of functions as an alternative reality, where objects are created and destroyed in a function call.

This is why you do not see hypotenuse listed in the environment - it has already been destroyed.

Global vs. Local Environments

Things can get confusing when you use the same names for variables both inside and outside a function. Check out this example:

```
pressure = 103.9
adjust <- function(t){
    temperature = t * 1.43 / pressure
    return(temperature)
}
pressure
#> [1] 104
temperature
#> Error in eval(expr, envir, enclos): object 'temperature' not found
```

`t` and `temperature` are **local** variables in `adjust`. Local variables are:

- Defined in the function.
- Not visible in the main program.
- Remember: a function parameter is a variable that is automatically assigned a value when the function is called.

`pressure` is a **global** variable. Global variables are:

- Defined outside any particular function.
- Visible everywhere.

This difference is referred to as **scope**. The **scope** of a variable is the part of a program that can 'see' that variable.

16.2.6 Arguments

Functions do not need to take input.

```
print_hello <- function(){
    print("hello")
}
print_hello()
#> [1] "hello"
```

But if a function takes input, arguments can be passed to functions in different ways.

1) **Positional arguments** are mandatory and have no default values.

```
send <- function(message, recipient){
  message <- paste(message, recipient)
  return(message)
}
send("Hello", "world")
#> [1] "Hello world"
```

In the case above, it is possible to use argument **names** when calling the functions and, by doing so, it is possible to switch the order of arguments. For instance:

```
send(recipient='World', message='Hello')
#> [1] "Hello World"
```

However, positional arguments (`send('Hello', 'World')`) are greatly perfered over names (`send(recipient='World', message='Hello')`), as it is very easy to accidentally specifying incorrect argument values.

2) **Keyword arguments** are not mandatory and have default values. They are often used for optional parameters sent to the function.

```
send <- function(message, recipient, cc=NULL){
  message <- paste(message, recipient, "cc:", cc)
  return(message)
}
send("Hello", "world")
#> [1] "Hello world cc: "
send("Hello", "world", "rochelle")
#> [1] "Hello world cc: rochelle"
```

Here `cc` and bcc are `optional`, and evaluate to `NULL` when they are not passed another value.

16.2.7 Challenges

Challenge 1

Write a function that calculates the sum of the squared value of two numbers. For instance, it should generate the following output:

```
my_function(3, 4)
# [1] 25
```

Challenge 2

Write `both_na()`, a function that takes two vectors of the same length and returns the number of positions that have an NA in both vectors.

Challenge 3

Fill in the blanks to create a function that takes a name like "Rochelle Terman" and returns that name in uppercase and reversed, like "TERMAN, ROCHELLE"

```
standard_names <- function(name){
    upper_case = toupper(____) # make upper
    upper_case_vec = strsplit(_____, split = ' ')[[1]] # turn into a vector
    first_name = ______ # take first name
    last_name = _______ # take last name
    reversed_name = paste(______, _______, sep = ", ") # reverse and separate by a comma and spac
    return(reversed_name)
}
```

Challenge 4

Look at the following function:

```
print_date <- function(year, month, day){
    joined = paste(as.character(year), as.character(month), as.character(day), sep = "/")
    return(joined)
}
```

What does this short program print?

```
print_date(day=1, month=2, year=2003)
```

Acknowledgements and Resources

- R for Data Science.
- Computing for Social Sciences

16.3 Iteration

In the last unit, we talked about how important it is to reduce duplication in your code by creating functions instead of copying-and-pasting. Avoiding duplication allows for more readable, more flexible, and less error-prone code.

Functions are one method of reducing duplication in your code. Another tool for reducing duplication is **iteration**, which lets you do the same task to multiple inputs.

In this chapter you'll learn about four approaches to iteratation:

1. Vectorized functions
2. For-loops
3. `map` and functional programming
4. Scoped verbs in `dplyr`

16.3.1 Vectorized Functions

Most of R's built-in functions are **vectorized**, meaning that the function will operate on all elements of a vector without needing to loop through and act on each element at a time.

That means you should never need to perform explicit iteration when performing simple mathematical computations.

```
x <- 1:4
x * 2
#> [1] 2 4 6 8
```

Notice that the multiplication happened to each element of the vector. Most built-in functions also operate element-wise on vectors:

```
x <- 1:4
log(x)
#> [1] 0.000 0.693 1.099 1.386
```

We can also add two vectors together:

```
x <- 1:4
y <- 6:9
```

```
x + y
#> [1]  7  9 11 13
```

Notice that each element of x was added to its corresponding element of y:

```
x:  1  2  3  4
    +  +  +  +
y:  6  7  8  9
---------------
    7  9 11 13
```

What happens if you add two vectors of different lengths?

```
1:10 + 1:2
#>  [1]  2  4  4  6  6  8  8 10 10 12
```

Here, R will expand the shortest vector to the same length as the longest. This is called **recycling**. This usually (but not always) happens silently, meaning R will not warn you. Beware!

16.3.2 For-loops

You will frequently need to iterate over vectors or data frames, perform an operation on each element, and save the results somewhere.

For example, imagine we have this simple dataframe:

```
df <- data.frame(
  a = rnorm(10),
  b = rnorm(10),
  c = rnorm(10),
  d = rnorm(10)
)
```

We want to compute the median of each column. You *could* do with copy-and-paste:

```
median(df$a)
#> [1] -0.246
median(df$b)
#> [1] -0.287
median(df$c)
#> [1] -0.0567
median(df$d)
#> [1] 0.144
```

But that breaks our rule of thumb: never copy and paste more than twice. Instead, we could use a `for` loop:

```
output <- vector("double", ncol(df))  # 1. output
for (i in seq_along(df)) {            # 2. sequence
  output[i] <- median(df[[i]])        # 3. body
}
output
#> [1] -0.2458 -0.2873 -0.0567  0.1443
```

Components of a `for` Loop

Every `for` loop has three components:

1. The **output**: `output <- vector("double", length(x))`.

Before you start a loop, you need to create an empty vector to store the output of the loop. Notice that the object is created **outside** the loop!

Preallocating space for your output is very important for efficiency: if you grow the `for` loop at each iteration using `c()` (for example), your `for` loop will be very slow.

2. The **sequence**: `i in seq_along(df)`.

This determines what to loop over. In this case, the sequence is `seq_along(df)`, which creates a numeric vector for a sequence of numbers beginning at 1 and continuing until it reaches the length of `df` (the length here is the number of columns in `df`).

```
seq_along(df)
#> [1] 1 2 3 4
```

It's useful to think of `i` as a pronoun, like "it". Each iteration of the `for` loop will assign `i` to a new value based on the designed sequence:

```
| Iteration | i =  |
|-----------|------|
| 1         | 1    |
| 2         | 2    |
| 3         | 3    |
| 4         | 4    |
=
```

NB: `seq_along` is a safe version of the more familiar `1:length(l)`, with an important difference: if you have a zero-length vector, `seq_along()` does the right thing:

```
y <- vector("double", 0)
seq_along(y)
#> integer(0)
```

```
1:length(y)
#> [1] 1 0
```

You probably won't create a zero-length vector deliberately, but it's easy to create one accidentally. If you use `1:length(x)` instead of `seq_along(x)`, you're likely to get a confusing error message.

3. The **body**: `output[[i]] <- median(df[[i]])`.

This is the code that does the work. It runs repeatedly, each time with a different value for `i`:

```
| Iteration | i =  | body                            |
|-----------|------|---------------------------------|
| 1         | 1    | output[[1]] <- median(df[[1]]) |
| 2         | 2    | output[[2]] <- median(df[[2]]) |
| 3         | 3    | output[[3]] <- median(df[[3]]) |
| 4         | 4    | output[[4]] <- median(df[[4]]) |
```

NB:: We use `[[` notation to reference each column of `df` using indices of columns, instead of `$` and column names.

16.3.3 Challenges

Challenge 1.

Fill in the blanks to write a `for` loop that calculates the arithmetic mean for every column in `mtcars`.

```
mtcars.means <- vector("double", ______)
for(i in ______){
  ______[i] <- mean(______[[i]])
}
```

Challenge 2.

Check out the `iris` dataset:

```
kable(head(iris))
```

Sepal.Length

Sepal.Width

Petal.Length

Petal.Width

Species

5.1

3.5

1.4

0.2

setosa

4.9

3.0

1.4

0.2

setosa

4.7

3.2

1.3

0.2

setosa

4.6

3.1

1.5

0.2

setosa

5.0

3.6

1.4

0.2

setosa

5.4

3.9

1.7

0.4

setosa

Write a `for` loop that calculates the number of unique values in each column of `iris`. Before you write the `for` loop, identify the three components you need:

1. Output
2. Sequence
3. Body

Challenge 3.

Generate 10 random normals for each of $\mu = -10$, 0, 10, and 100. Store them in a list.

16.3.4 Functional Programming and `map`

Loops are not as important in R as they are in other languages because R is a **functional** programming language. This means that it's possible to wrap up `for` loops in a function, and call that function instead of using the for loop directly.

The pattern of looping over a vector, doing something to each element and saving the results is so common that the `purrr` package provides a family of functions to do it for you. They effectively eliminate the need for many common `for` loops.

There is one function for each type of output:

1. `map()` makes a list.
2. `map_lgl()` makes a logical vector.
3. `map_int()` makes an integer vector.
4. `map_dbl()` makes a double vector.
5. `map_chr()` makes a character vector.

Each function takes a vector as input, applies a function to each piece, and then returns a new vector that's the same length (and has the same names) as the input.

NB: Some people will tell you to avoid `for` loops because they are slow. They're wrong! (Well, at least they're rather out of date, as for loops haven't been slow for many years). The main benefit of using functions like `map()` is not speed, but clarity: they make your code easier to write and to read.

To see how `map` works, consider (again) this simple data frame:

```
df <- tibble(
  a = rnorm(10),
  b = rnorm(10),
  c = rnorm(10),
```

```
  d = rnorm(10)
)
```

What if we wanted to calculate the mean, median, and standard deviation of each column?

```
map_dbl(df, mean)
#>      a      b      c      d 
#>  0.116  0.127 -0.089  0.281
map_dbl(df, median)
#>       a       b       c       d 
#>  0.0583  0.0244 -0.0571  0.2604
map_dbl(df, sd)
#>     a     b     c     d 
#> 1.161 1.226 1.024 0.798
```

Compared to using a for loop, this approach is much easier to read, and less error-prone.

The data can even be piped!

```
df %>% map_dbl(mean)
#>      a      b      c      d 
#>  0.116  0.127 -0.089  0.281
df %>% map_dbl(median)
#>       a       b       c       d 
#>  0.0583  0.0244 -0.0571  0.2604
df %>% map_dbl(sd)
#>     a     b     c     d 
#> 1.161 1.226 1.024 0.798
```

We can also pass additional arguments. For example, the function `mean` passes an optional argument `trim`. From the help file: "the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the `mean`is computed.

```
map_dbl(df, mean, trim = 0.5)
#>       a       b       c       d 
#>  0.0583  0.0244 -0.0571  0.2604
```

Check out other fun applications of `map` functions here

16.3.5 Challenges

Write code that uses one of the map functions to:

1. Calculates the arithmetic mean for every column in `mtcars`.
2. Calculates the number of unique values in each column of `iris`.

3. Generate 10 random normals for each of $\mu = -10$, 0, 10, and 100.

16.3.6 Scoped Verbs

The last iteration technique we'll discuss is scoped verbs in `dplyr`.

Frequently, when working with dataframes, we want to apply a function to multiple columns. For example, let's say we want to calculate the mean value of each column in `mtcars`.

If we wanted to calculate the average of a single column, it would be pretty simple using just regular `dplyr` functions:

```
mtcars %>%
  summarize(mpg = mean(mpg))
#>    mpg
#> 1 20.1
```

But if we want to calculate the mean for all of them, we'd have to duplicate this code many times over:

```
mtcars %>%
  summarize(mpg = mean(mpg),
            cyl = mean(cyl),
            disp = mean(disp),
            hp = mean(hp),
            drat = mean(drat),
            wt = mean(wt),
            qsec = mean(qsec),
            vs = mean(vs),
            am = mean(am),
            gear = mean(gear),
            carb = mean(carb))
#>    mpg  cyl disp  hp drat   wt qsec    vs    am gear carb
#> 1 20.1 6.19  231 147  3.6 3.22 17.8 0.438 0.406 3.69 2.81
```

This is very repetitive and prone to mistakes!

We just saw one approach to solve this problem: `map`. Another approach is **scoped verbs**.

Scoped verbs allow you to use standard verbs (or functions) in `dplyr` that affect multiple variables at once.

- `_if` allows you to pick variables based on a predicate function like `is.numeric()` or `is.character()`
- `_at` allows you to pick variables using the same syntax as `select()`
- `_all` operates on all variables

These verbs can apply to `summarize`, `filter`, or `mutate`. Let's go over `summarize`:

`summarize_all()`

`summarize_all()` takes a dataframe and a function and applies that function to each column:

```
mtcars %>%
  summarize_all(.funs = mean)
#>    mpg  cyl disp  hp drat   wt qsec    vs    am gear carb
#> 1 20.1 6.19  231 147  3.6 3.22 17.8 0.438 0.406 3.69 2.81
```

`summarize_at()`

`summarize_at()` allows you to pick columns in the same way as `select()`, that is, based on their names. There is one small difference: you need to wrap the complete selection with the `vars()` helper (this avoids ambiguity).

```
mtcars %>%
  summarize_at(.vars = vars(mpg, wt), .funs = mean)
#>    mpg   wt
#> 1 20.1 3.22
```

`summarize_if()`

`summarize_if()` allows you to pick variables to summarize based on some property of the column. For example, what if we want to apply a numeric summary function only to numeric columns:

```
iris %>%
  summarize_if(.predicate = is.numeric, .funs = mean)
#>   Sepal.Length Sepal.Width Petal.Length Petal.Width
#> 1         5.84        3.06         3.76         1.2
```

`mutate` and `filter` work in a similar way. To see more, check out Scoped verbs by the Data Challenge Lab

Acknowledgments

A good portion of this lesson is based on:

- R for Data Science
- Computing for Social Sciences

Chapter 17

Collecting Data from the Web

17.1 Introduction

There's a ton of web data that's useful to social scientists, including:

- social media
- news media
- government publications
- organizational records

There are two ways to get data off the web:

1. **Web APIs** - i.e. application-facing, for computers
2. **Webscraping** - i.e. user-facing websites for humans

Rule of Thumb: Check for API first. If not available, scrape.

17.2 Web APIs

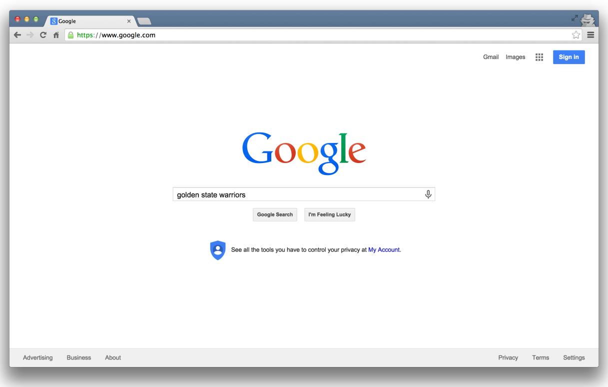
API stands for **Application Programming Interface**. Broadly defined, an API is a set of rules and procedures that facilitate interactions between computers and their applications.

A very common type of API is the **Web API**, which (among other things) allows users to query a remote database over the internet.

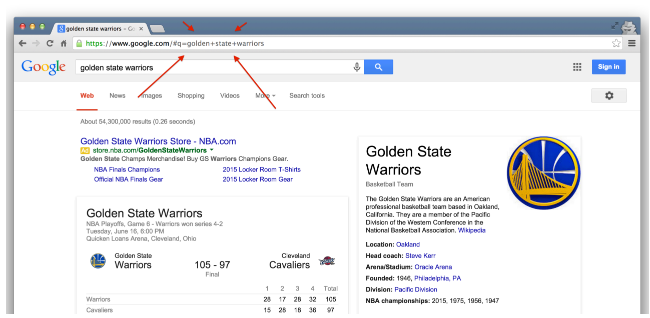
Web APIs take on a variety of formats, but the vast majority adhere to a particular style known as **Reperesentational State Transfer** or **REST**. What makes these "RESTful" APIs so convenient is that we can use them to query databases using URLs.

RESTful Web APIs are All Around You...

Consider a simple Google search:



Ever wonder what all that extra stuff in the address bar was all about? In this case, the full address is Google's way of sending a query to its databases requesting information related to the search term "golden state warriors".



In fact, it looks like Google makes its query by taking the search terms, separating each of them with a "+", and appending them to the link "https://www.google.com/#q=". Therefore, we should be able to actually change our Google search by adding some terms to the URL and following the general format...

Learning how to use RESTful APIs is all about learning how to format these URLs so that you can get the response you want.

17.2.1 Some Basic Terminology

Let's get on the same page with some basic terminology:

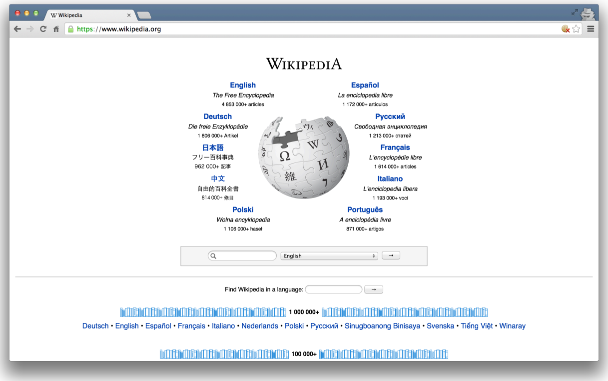
- **Uniform Resource Location (URL)**: a string of characters that, when interpreted via the Hypertext Transfer Protocol (HTTP), points to a data resource, notably files written in Hypertext Markup Language (HTML) or a subset of a database. This is often referred to as a "call".
- **HTTP Methods/Verbs**:
  - *GET*: requests a representation of a data resource corresponding to a particular URL. The process of executing the GET method is often referred to as a "GET request" and is the main method used for querying RESTful databases.
  - *HEAD*, *POST*, *PUT*, *DELETE*: other common methods, though mostly never used for database querying.

17.2.2 How Do GET Requests Work?

A Web Browsing Example

As you might suspect from the example above, surfing the web is basically equivalent to sending a bunch of `GET` requests to different servers and asking for different files written in HTML.

Suppose, for instance, I wanted to look something up on Wikipedia. My first step would be to open my web browser and type in http://www.wikipedia.org. Once I hit return, I'd see the page below.



Several different processes occured, however, between me hitting "return" and the page finally being rendered. In order:

1. The web browser took the entered character string and used the command-line tool "Curl" to write a properly formatted HTTP GET request and submitted it to the server that hosts the Wikipedia homepage.
2. After receiving this request, the server sent an HTTP response, from which Curl extracted the HTML code for the page (partially shown below).
3. The raw HTML code was parsed and then executed by the web browser, rendering the page as seen in the window.

```
#> No encoding supplied: defaulting to UTF-8.
#> [1] "<!DOCTYPE html>\n<html lang=\"mul\" class=\"no-js\">\n<head>\n<meta charset=\"
```

Web Browsing as a Template for RESTful Database Querying

The process of web browsing described above is a close analogue for the process of database querying via RESTful APIs, with only a few adjustments:

1. While the Curl tool will still be used to send HTML GET requests to the servers hosting our databases of interest, the character string that we supply to Curl must be constructed so that the resulting request can be interpreted and succesfully acted upon by the server. In particular, it is likely that the character string must encode **search terms and/or filtering parameters**, as well as one or more **authentication codes**. While the terms are often similar across APIs, most are API-specific.
2. Unlike with web browsing, the content of the server's response that is extracted by Curl is unlikely to be HTML code. Rather, it will likely be **raw text response that can be parsed into one of a few file formats commonly used for data storage**. The usual suspects include .csv, .xml, and .json files.

3. Whereas the web browser capably parsed and executed the HTML code, **one or more facilities in R, Python, or other programming languages will be necessary for parsing the server response and converting it into a format for local storage** (e.g. matrices, dataframes, databases, lists, etc.).

17.2.3 Finding APIs

More and more APIs pop up every day. Programmable Web offers a running list of APIs. This list provides a list of APIs that may be useful to Political Scientists.

Here are some APIs that may be useful to you:

- NYT Article API: Provides metdata (title, summaries, dates, etc) from all New York Times articles in their archive.
- GeoNames geographical database: Provides lots of geographical information for all countries and other locations. The `geonames` package provides a wrapper for R.
- The Manifesto Project: Provides text and other information on political party manifestos from around the world. It currently covers over 1000 parties from 1945 until today in over 50 countries on five continents. The `manifestoR` package provides a wrapper for R.
- The Census Bureau: Provides datasets from US Census Bureau. The `tidycensus` package allows users to interface with the US Census Bureau's decennial Census and five-year American Community APIs.

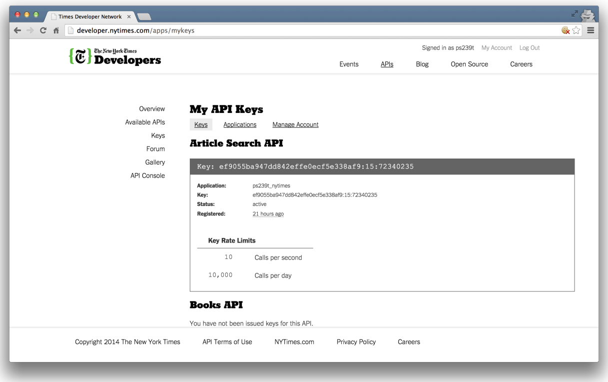
17.2.4 Getting API Access

Most APIs requires a key or other user credentials before you can query their database.

Getting credentialized with a API requires that you register with the organization. Most APIs are set up for developers, so you'll likely be asked to register an "application". All this really entails is coming up with a name for your app/bot/project, and providing your real name, organization, and email. Note that some more popular APIs (e.g. Twitter, Facebook) will require additional information, such as a web address or mobile number.

Once you've successfully registered, you will be assigned one or more keys, tokens, or other credentials that must be supplied to the server as part of any API call you make. To make sure that users aren't abusing their data access privileges (e.g. by making many rapid queries), each set of keys will be given **rate limits** governing the total number of calls that can be made over certain intervals of time.

For example, the NYT Article API has relatively generous rate limits — 4,000 requests per day and 10 requests per minute. So we need to "sleep"" 6 seconds between calls to avoid hitting the per minute rate limit.



17.2.5 Using APIs in R

There are two ways to collect data through APIs in R.

1. **Plug-n-play packages**

Many common APIs are available through user-written R Packages. These packages offer functions that "wrap" API queries and format the response. These packages are usually much more convenient than writing our own query, so it's worth searching around for a package that works with the API we need.

2. **Writing our own API request**

If no wrapper function is available, we have to write our own API request, and format the response ourselves using R. This is trickier, but definitely do-able.

17.3 Collecting Twitter Data with RTweet

Twitter actually has two separate APIs:

1. The **REST API** allows you to read and write Twitter data. For research purposes, this allows you to search the recent history of tweets and look up specific users.
2. The **Streaming API** allows you to access public data flowing through Twitter in real-time. It requires your R session to be running continuously, but allows you to capture a much larger sample of tweets while avoiding rate limits for the REST API.

There are several packages for R for accessing and searching Twitter. In this unit, we'll practice using the `RTweet` library, which allows us to easily collect data from Twitter's REST and stream APIs.

17.3.1 Setting up `RTweet`

To use `RTweet`, follow these steps:

1. If you don't have a Twitter account, create one here.
2. Install the `RTweet` package from CRAN.
3. Load the package into R.
4. Send a request to Twitter's API by calling any of the package's functions, like `search_tweets` or`get_timeline`.
5. Approve the browser popup (to authorize the `rstats2twitter` app).
6. Now, you're ready to use RTweet!

Let's go ahead and load `RTweet` along with some other helpful functions:

```
library(tidyverse)
library(rtweet)
library(lubridate)
library(kableExtra)
```

17.3.2 UChicago Political Science Prof Tweets

Let's explore the `RTweet` package to see what we can learn about the tweeting habits of UChicago Political Science faculty.

The function `get_timeline` will pull the most recent `n` number of tweets from a given handle(s). To pull tweets from multiple handles, write out a vector of the handles in the `user` argument.

Let's pull tweets from five faculty members in the department.

```
profs <- get_timeline(
  user = c("carsonaust", "profpaulpoast", "pstanpolitics", "rochelleterman", "bobbygulotty"),
  n = 1000
)
kable(head(profs))
```

user_id

status_id

created_at

screen_name

text

source

display_text_width

reply_to_status_id

reply_to_user_id

reply_to_screen_name

is_quote

is_retweet

favorite_count

retweet_count

quote_count

reply_count

hashtags

symbols

urls_url

urls_t.co

urls_expanded_url

media_url

media_t.co

media_expanded_url

media_type

ext_media_url

ext_media_t.co

ext_media_expanded_url

ext_media_type

mentions_user_id

mentions_screen_name

lang

quoted_status_id

quoted_text

quoted_created_at

quoted_source

quoted_favorite_count

quoted_retweet_count

quoted_user_id

quoted_screen_name

quoted_name

quoted_followers_count

quoted_friends_count

quoted_statuses_count

quoted_location

quoted_description

quoted_verified

retweet_status_id

retweet_text

retweet_created_at

retweet_source

retweet_favorite_count

retweet_retweet_count

retweet_user_id

retweet_screen_name

retweet_name

retweet_followers_count

retweet_friends_count

retweet_statuses_count

retweet_location

retweet_description

retweet_verified

place_url

place_name

place_full_name

place_type

country

country_code

geo_coords

coords_coords

bbox_coords

status_url

name

location

description

url

protected

followers_count

friends_count

listed_count

statuses_count

favourites_count

account_created_at

verified

profile_url

profile_expanded_url

account_lang

profile_banner_url

profile_background_url

profile_image_url

805833715

1193167372481417216

1573308244

carsonaust

? ? ? ? ? ? ? ? ? ? ...and so many more

Twitter Web App

149

1193167371437039617

805833715

carsonaust

FALSE

FALSE

4

0

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

c("1036404458161426432", "1636747982", "821082773884637184", "256659585", "2618035261", "297648980", "718793610", "55015468", "2364189422", "2826743402")

c("ProfTalmadge", "ProfSaunders", "ANewman_forward", "Kat_McNamara", "IRgetsreal", "dbyman", "dmedelstein", "dhnexon", "ErikVoeten", "irfannoorud-din")

en

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

c(NA, NA)

c(NA, NA)

c(NA, NA, NA, NA, NA, NA, NA, NA)

https://twitter.com/carsonaust/status/1193167372481417216

Austin Carson

Chicago, IL

Assistant Professor @ University of Chicago, author of *Secret Wars: Covert Conflict in International Politics*. Dad & NBA / Pistons enthusiast

https://t.co/kApIyoe7RG

FALSE

2003

1026

24

1542

2683

1346896669

FALSE

https://t.co/kApIyoe7RG

https://austinmcarson.com/

NA

https://pbs.twimg.com/profile_banners/805833715/1533067436

http://abs.twimg.com/images/themes/theme1/bg.png

http://pbs.twimg.com/profile_images/1024384740181336064/ROqTG_uN_normal.jpg

805833715

1193167371437039617

1573308243

carsonaust

Really excited to head to DC for this event on Monday **?**. Truly honored to be receiving the Lepgold Book Prize for *Secret Wars*. And bonus! A chance to see some of the many amazing folks at Georgetown https://t.co/DpBEozAJQR

Twitter Web App

237

NA

NA

NA

FALSE

FALSE

103

6

NA

NA

NA

NA

mortara.georgetown.edu/event/lecture-...

https://t.co/DpBEozAJQR

https://mortara.georgetown.edu/event/lecture-by-lepgold-book-prize-winner-austin-carson/

NA

NA

NA

NA

NA

NA

NA

NA

48739999

MortaraCenter

en

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

c(NA, NA)

c(NA, NA)

c(NA, NA, NA, NA, NA, NA, NA, NA)

https://twitter.com/carsonaust/status/1193167371437039617

Austin Carson

Chicago, IL

Assistant Professor @ University of Chicago, author of *Secret Wars: Covert Conflict in International Politics*. Dad & NBA / Pistons enthusiast

https://t.co/kApIyoe7RG

FALSE

2003

1026

24

1542

2683

1346896669

FALSE

https://t.co/kApIyoe7RG

https://austinmcarson.com/

NA

https://pbs.twimg.com/profile_banners/805833715/1533067436

http://abs.twimg.com/images/themes/theme1/bg.png

http://pbs.twimg.com/profile_images/1024384740181336064/ROqTG_uN_normal.jpg

805833715

1192955815625469953

1573257805

carsonaust

All of you are too kind! **?** was quick on the draw. Am all set, thank you!

Twitter for iPhone

80

1192938163993493509

805833715

carsonaust

FALSE

FALSE

2

0

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

1702865540

mevers90

en

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

c(NA, NA)

c(NA, NA)

c(NA, NA, NA, NA, NA, NA, NA, NA)

https://twitter.com/carsonaust/status/1192955815625469953

Austin Carson

Chicago, IL

Assistant Professor @ University of Chicago, author of *Secret Wars: Covert Conflict in International Politics*. Dad & NBA / Pistons enthusiast

https://t.co/kApIyoe7RG

FALSE

2003

1026

24

1542

2683

1346896669

FALSE

https://t.co/kApIyoe7RG

https://austinmcarson.com/

NA

https://pbs.twimg.com/profile_banners/805833715/1533067436

http://abs.twimg.com/images/themes/theme1/bg.png

http://pbs.twimg.com/profile_images/1024384740181336064/ROqTG_uN_normal.jpg

805833715

1192938163993493509

1573253596

carsonaust

Anyone have access to this?

Mapping fears: the use of commercial high-resolution satellite imagery in international affairs S Livingston, WL Robinson - Astropolitics, 2003 https://t.co/CbP7h6qG1G

Twitter Web App

198

NA

NA

NA

FALSE

FALSE

3

1

NA

NA

NA

NA

tandfonline.com/doi/abs/10.108…

https://t.co/CbP7h6qG1G

https://www.tandfonline.com/doi/abs/10.1080/14777620312331269909

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

en

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

c(NA, NA)

c(NA, NA)

c(NA, NA, NA, NA, NA, NA, NA, NA)

https://twitter.com/carsonaust/status/1192938163993493509

Austin Carson

Chicago, IL

Assistant Professor @ University of Chicago, author of *Secret Wars: Covert Conflict in International Politics*. Dad & NBA / Pistons enthusiast

https://t.co/kApIyoe7RG

FALSE

2003

1026

24

1542

2683

1346896669

FALSE

https://t.co/kApIyoe7RG

https://austinmcarson.com/

NA

https://pbs.twimg.com/profile_banners/805833715/1533067436

http://abs.twimg.com/images/themes/theme1/bg.png

http://pbs.twimg.com/profile_images/1024384740181336064/ROqTG_uN_normal.jpg

805833715

1192858707253702657

1573234652

carsonaust

? ? No that's plagiarism. An F!

Twitter Web App

27

1192857507699539971

702215720

na_paraiso

FALSE

FALSE

8

0

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

c("702215720", "1001654539752271874")

c("na_paraiso", "krasnican_")

en

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

c(NA, NA)

c(NA, NA)

c(NA, NA, NA, NA, NA, NA, NA, NA)

https://twitter.com/carsonaust/status/1192858707253702657

Austin Carson

Chicago, IL

Assistant Professor @ University of Chicago, author of *Secret Wars: Covert Conflict in International Politics*. Dad & NBA / Pistons enthusiast

https://t.co/kApIyoe7RG

FALSE

2003

1026

24

1542

2683

1346896669

FALSE

https://t.co/kApIyoe7RG

https://austinmcarson.com/

NA

https://pbs.twimg.com/profile_banners/805833715/1533067436

http://abs.twimg.com/images/themes/theme1/bg.png

http://pbs.twimg.com/profile_images/1024384740181336064/ROqTG_uN_normal.jpg

805833715

1192855989902618624

1573234004

carsonaust

Well played, sir!

<opens gradebook> <goes to “Krasnican> <enters”A+”> https://t.co/fbOPeJpcVM

Twitter Web App

89

NA

NA

NA

TRUE

FALSE

36

0

NA

NA

NA

NA

twitter.com/krasnican_/sta…

https://t.co/fbOPeJpcVM

https://twitter.com/krasnican_/status/1192491736670625793

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

en

1192491736670625793

PSA: We're halfway through the quarter **?** and this course I'm taking called "Becoming a Global Power: The American Experience" is without a doubt the most interesting class I have ever taken. Taught by **?**. Everyone should be jealous

1573147160

Twitter Web App

18

0

1001654539752271874

krasnican_

Thomas Krasnican

213

236

447

Anchorage - Annapolis - Chicago

MPP '20 ? | USNA '18 | Director of ? | 2019 Military & NatSec Fellow ? | Tweets endorsement by DoD

FALSE

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

NA

c(NA, NA)

c(NA, NA)

c(NA, NA, NA, NA, NA, NA, NA, NA)

https://twitter.com/carsonaust/status/1192855989902618624

Austin Carson

Chicago, IL

Assistant Professor @ University of Chicago, author of *Secret Wars: Covert Conflict in International Politics*. Dad & NBA / Pistons enthusiast

https://t.co/kApIyoe7RG

FALSE

2003

1026

24

1542

2683

1346896669

FALSE

https://t.co/kApIyoe7RG

https://austinmcarson.com/

NA

https://pbs.twimg.com/profile_banners/805833715/1533067436

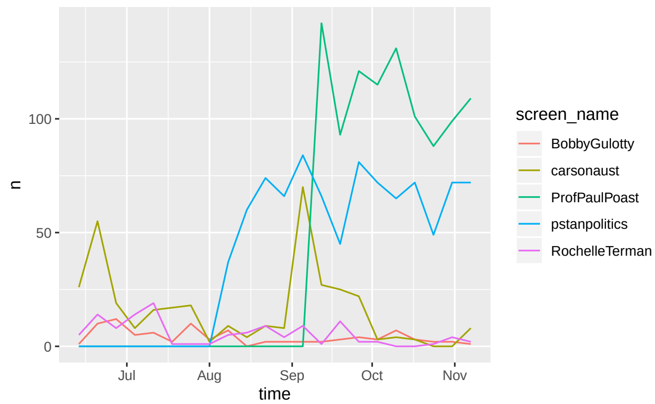
http://abs.twimg.com/images/themes/theme1/bg.png

http://pbs.twimg.com/profile_images/1024384740181336064/ROqTG_uN_normal.jpg

Now, let's visualize which professors are tweeting the most, by week.

```
profs %>%
  group_by(screen_name) %>%
  mutate(created_at = as.Date(created_at)) %>%
```

```
  filter(created_at >= "2019-06-15") %>%
  ts_plot(by = "week")
```



17.3.3 Hashtags and Text Strings

We can also use `RTweet` to explore certain hashtags or text strings.

Let's take Duke Ellington again – we can use `search_tweets` to pull the most recent n number of tweets that include the hashtag `#DukeEllington` or the string `"Duke Ellington"`.

Hashtag Challenge

Using the documentation for `search_tweets` as a guide, try pulling the 2,000 most recent tweets that include `#DukeEllington` or `"Duke Ellington"` – be sure to exclude retweets from the query.

1. Why didn't your query return 2,000 results?
2. Identify the user that has used either the hashtag or the string in the greatest number of tweets – where is this user from?

```
duke <- search_tweets(
  q = '#DukeEllington OR "Duke Ellington"',
  n = 2000,
  include_rts = FALSE
)

duke %>%
  group_by(user_id, location) %>%
  summarise(n = n()) %>%
  arrange(desc(n))
```

```
#> # A tibble: 712 x 3
#> # Groups:   user_id [712]
#>   user_id             location                           n
#>   <chr>               <chr>                          <int>
#> 1 2561373848          Florida, USA                      84
#> 2 764290456599396353  Rochester, New York               39
#> 3 1022122568004894720 ""                                31
#> 4 1030069194539368448 ""                                30
#> 5 714738130557992960  Chiclana de la Frontera, Spain    24
#> 6 826869114           Paris, Ile-de-France              18
#> # ... with 706 more rows
```

17.4 Writing API Queries

If no wrapper package is available, we have to write our own API query, and format the response ourslves using R. This is trickier, but definitely do-able.

In this unit, we'll practice constructing our own API queries using the New York Time's `Article API`. This API provides metadata (title, date, summary, etc) on all of New York Times articles.

Fortunately, this API is very well documented!

You can even try it out here.

Load the following packages to get started:

```
library(tidyverse)
library(httr)
library(jsonlite)
library(lubridate)
```

17.4.1 Constructing the API GET Request

Likely the most challenging part of using web APIs is learning how to format your GET request URLs. While there are common architectures for such URLs, each API has its own unique quirks. For this reason, carefully reviewing the API documentation is critical.

Most GET request URLs for API querying have three or four components:

1. *Authenication Key/Token*: a user-specific character string appended to a base URL telling the server who is making the query; allows servers to efficiently manage database access

2. *Base URL*: a link stub that will be at the beginning of all calls to a given API; points the server to the location of an entire database
3. *Search Parameters*: a character string appended to a base URL that tells the server what to extract from the database; basically a series of filters used to point to specific parts of a database
4. *Response Format*: a character string indicating how the response should be formatted; usually one of .csv, .json, or .xml

Let's go ahead and store the these values as variables:

```
key <- "Onz0BobMTn2IRJ7krcT5RXHknkGLqiaI"
base.url <- "http://api.nytimes.com/svc/search/v2/articlesearch.json"
search_term <- "John Mearsheimer"
```

How did I know the `base.url`? I read the documentation.. Notice that this `base.url` also includes the *response format*(`.jston`), so we don't need to configure that directly.

We're ready to make the request. We can use the `GET` function from the `httr` package (another `tidyverse` package) to make an HTTP GET Request.

```
r <- GET(base.url, query = list(`q` = search_term,
                                `api-key` = key))
```

Now, we have an object called `r`. We can get all the information we need from this object. For instance, we can see that the URL has been correctly encoded by printing the URL. Click on the link to see what happens.

```
r$url
#> [1] "http://api.nytimes.com/svc/search/v2/articlesearch.json?q=John%20Mearsheimer&api-key=Onz0
```

Challenge 1: Adding a date range

What if we only want to search within a particular date range? The NYT Article Api allows us to specify start and end dates.

Alter the `get.request` code above so that the request only searches for articles in the year 2005.

You're gonna need to look at the documentation here to see how to do this.

Challenge 2: Specifying a results page

The above will return the first 10 results. To get the next ten, you need to add a "page" parameter. Change the search parameters above to get the second 10 resuls.

17.4.2 Parsing the response

We can read the content of the server's response using the `content()` function.

```
response <- content(r, "text")
substr(response, start = 1, stop = 1000)
#> [1] "{\"status\":\"OK\",\"copyright\":\"Copyright (c) 2019 The New York Times Compa
```

What you see here is JSON text, encoded as plain text. JSON stands for "Javascript object notation." Think of JSON like a nested array built on key/value pairs.

We want to convert the results from JSON format to something easier to work with – notably a data frame.

The **jsonlite** package provides several easy conversion functions for moving between JSON and vectors, data.frames, and lists. Let's use the function `fromJSON` to convert this response into something we can work with:

```
# Convert JSON response to a dataframe
response_df <- fromJSON(response, simplifyDataFrame = TRUE, flatten = TRUE)

# Inspect the dataframe
str(response_df, max.level = 2)
#> List of 3
#>  $ status   : chr "OK"
#>  $ copyright: chr "Copyright (c) 2019 The New York Times Company. All Rights Reserv
#>  $ response :List of 2
#>   ..$ docs:'data.frame': 10 obs. of  26 variables:
#>   ..$ meta:List of 3
```

That looks intimidating! But it's really just a big, nested list. Let's see what we got in there.

```
names(response_df)
#> [1] "status"    "copyright" "response"

# This is boring
response_df$status
#> [1] "OK"

# So is this
response_df$copyright
#> [1] "Copyright (c) 2019 The New York Times Company. All Rights Reserved."

# This is what we want!
names(response_df$response)
#> [1] "docs" "meta"
```

Within `response_df$response`, we can extract a number of interesting results, including the number of total hits, as well as information on the first ten documents:

```
# What's in 'meta'?
response_df$response$meta
#> $hits
#> [1] 168
#>
#> $offset
#> [1] 0
#>
#> $time
#> [1] 202

# pull out number of hits
response_df$response$meta$hits
#> [1] 168

# Check out docs
names(response_df$response$docs)
#>  [1] "web_url"                  "snippet"
#>  [3] "lead_paragraph"           "abstract"
#>  [5] "print_page"               "source"
#>  [7] "multimedia"               "keywords"
#>  [9] "pub_date"                 "document_type"
#> [11] "news_desk"                "section_name"
#> [13] "type_of_material"         "_id"
#> [15] "word_count"               "uri"
#> [17] "headline.main"            "headline.kicker"
#> [19] "headline.content_kicker" "headline.print_headline"
#> [21] "headline.name"            "headline.seo"
#> [23] "headline.sub"             "byline.original"
#> [25] "byline.person"            "byline.organization"

# put it in another variable
docs <- response_df$response$docs
```

17.4.3 Iteration through results pager

That's great. But we only have 10 items. The original response said we had 168 hits! Which means we have to make 168 /10, or 17 requests to get them all. Sounds like a job for iteration!

First, let's write a function that passes a search term and a page number, and returns a dataframe of articles.

```
nytapi <- function(term = NULL, n){
    base.url = "http://api.nytimes.com/svc/search/v2/articlesearch.json"
    key = "Onz0BobMTn2IRJ7krcT5RXHknkGLqiaI"

    # Send GET request
    r <- GET(base.url, query = list(`q` = term,
                                  `api-key` = key,
                                  `page` = n))

    # Parse response to JSON
    response <- content(r, "text")
    response_df <- fromJSON(response, simplifyDataFrame = T, flatten = T)

    print(paste("Scraping page: ", as.character(n)))

    return(response_df$response$docs)
}

docs <- nytapi("John Mearsheimer", 2)
#> [1] "Scraping page:  2"
```

Now, we're ready to iterate over each page. First, let's review what've done so far:

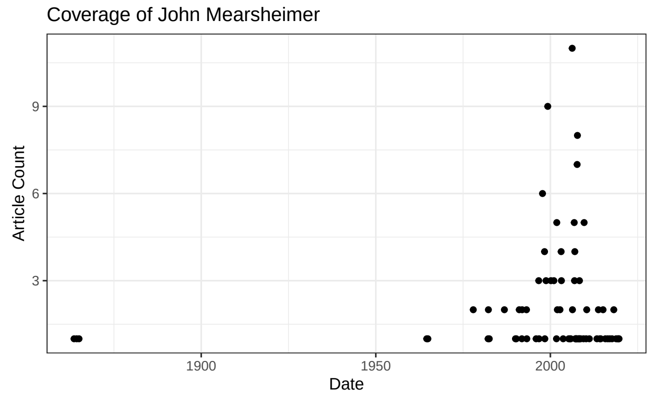
```
# set key and base
base.url = "http://api.nytimes.com/svc/search/v2/articlesearch.json"
key = "Onz0BobMTn2IRJ7krcT5RXHknkGLqiaI"
search_term = "John Mearsheimer" #change me

# Send GET request
r <- GET(base.url, query = list(`q` = search_term,
                                `api-key` = key))

# Parse response to JSON
response <- content(r, "text")
response_df <- fromJSON(response, simplifyDataFrame = T, flatten = T)

# extract hits
hits = response_df$response$meta$hits

# get number of pages
pages = ceiling(hits/10)

# modify function to sleep
nytapi_slow <- slowly(nytapi, rate = rate_delay(6))
```

```r
# iterate over pages, getting all docs
docs_list <- map((1:pages), ~nytapi_slow(term = search_term, n = .))
#> [1] "Scraping page:  1"
#> [1] "Scraping page:  2"
#> [1] "Scraping page:  3"
#> [1] "Scraping page:  4"
#> [1] "Scraping page:  5"
#> [1] "Scraping page:  6"
#> [1] "Scraping page:  7"
#> [1] "Scraping page:  8"
#> [1] "Scraping page:  9"
#> [1] "Scraping page:  10"
#> [1] "Scraping page:  11"
#> [1] "Scraping page:  12"
#> [1] "Scraping page:  13"
#> [1] "Scraping page:  14"
#> [1] "Scraping page:  15"
#> [1] "Scraping page:  16"
#> [1] "Scraping page:  17"

# flatten to create one bit dataframe
docs_df <- bind_rows(docs_list)
```

17.4.4 Visualizing Results

To figure out how John Mearsheimer's popularity is changing over time, all we need to do is add an indicator for the year and month each article was published in.

```r
# Format pub_date using lubridate
docs_df$date <- ymd_hms(docs_df$pub_date)

by_month <- docs_df %>% group_by(floor_date(date, "month")) %>%
  summarise(count  = n()) %>%
  rename(month = 1)

by_month %>%
  ggplot(aes(x = month, y = count)) +
  geom_point() +
  theme_bw() +
  xlab(label = "Date") +
  ylab(label = "Article Count") +
  ggtitle(label = "Coverage of John Mearsheimer")
```



17.4.5 More resources

The documentation for httr includes two useful vignettes:

1. httr quickstart guide - summarizes all the basic httr functions like above
2. Best practices for writing an API package - document outlining the key issues involved in writing API wrappers in R

17.5 Webscraping

If no API is available, we can scrape a website directory. Webscraping has a number of benefits and challenges compared to APIs:

Webscraping Benefits

- Any content that can be viewed on a webpage can be scraped. Period
- No API needed
- No rate-limiting or authentication (usually)

Webscraping Challenges

- Rarely tailored for researchers
- Messy, unstructured, inconsistent
- Entirely site-dependent

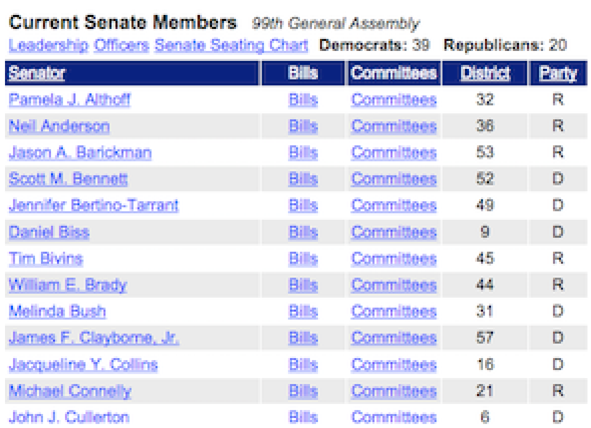
Some Disclaimers

- Check a site's terms and conditions before scraping.
- Be nice - don't hammer the site's server. Review these ethical webscraping tips
- Sites change their layout all the time. Your scraper will break.

17.5.1 What's a website?

A website is some combination of codebase and database. The "front end" product is HTML + CSS stylesheets + javascript, looking something like this:

Your browser turns that into a nice layout.



Current Senate Members *99th General Assembly*

Leadership Officers Senate Seating Chart **Democrats:** 39 **Republicans:** 20

Senator	Bills	Committees	District	Party
Pamela J. Althoff	Bills	Committees	32	R
Neil Anderson	Bills	Committees	36	R
Jason A. Barickman	Bills	Committees	53	R
Scott M. Bennett	Bills	Committees	52	D
Jennifer Bertino-Tarrant	Bills	Committees	49	D
Daniel Biss	Bills	Committees	9	D
Tim Bivins	Bills	Committees	45	R
William E. Brady	Bills	Committees	44	R
Melinda Bush	Bills	Committees	31	D
James F. Clayborne, Jr.	Bills	Committees	57	D
Jacqueline Y. Collins	Bills	Committees	16	D
Michael Connelly	Bills	Committees	21	R
John J. Cullerton	Bills	Committees	6	D

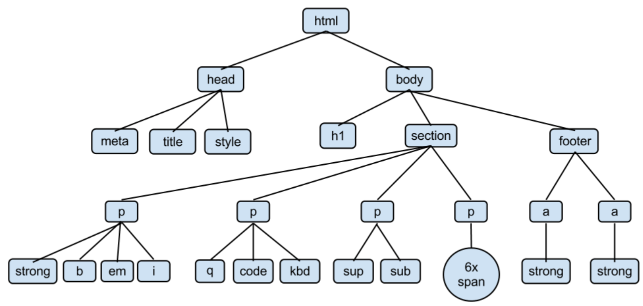
17.5.2 HTML

The core of a website is **HTML** (Hyper Text Markup Language.) HTML is composed of a tree of HTML _nodes**elements**, such as headers, paragraphs, etc.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Page title</title>
    </head>
    <body>
        <p>Hello world!</p>
```

```
    </body>
</html>
```

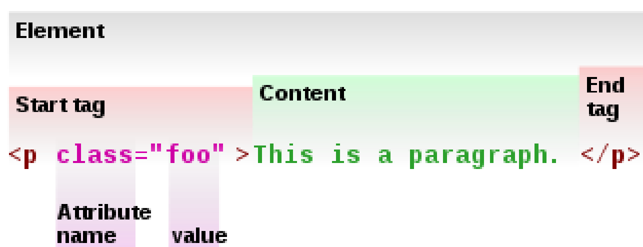
HTML elements can contain other elements:



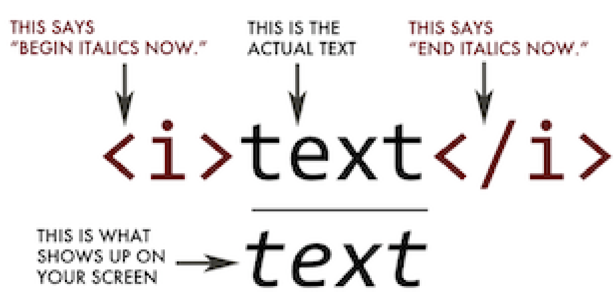
Generally speaking, an HTML element has three components:

1. Tags (starting and ending the element)
2. Atributes (giving information about the element)
3. Text, or Content (the text inside the element)

```
knitr::include_graphics(path = "img/html-element.png")
```



HTML: Tags

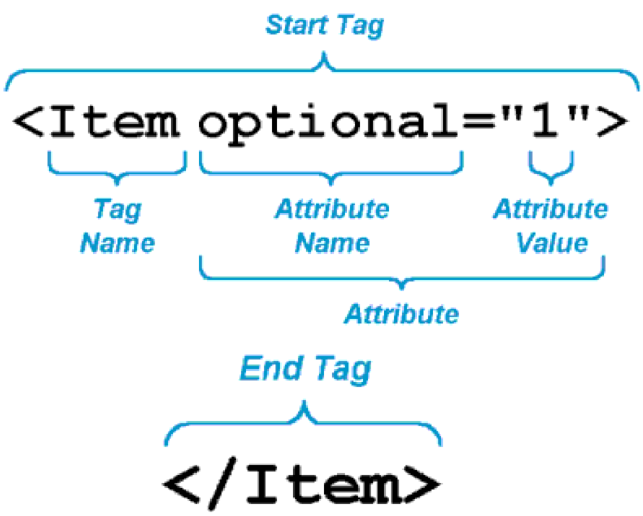


Common HTML tags

Tag	Meaning
<head>	page header (metadata, etc
<body>	holds all of the content
<p>	regular text (paragraph)
<h1>,<h2>,<h3>	header text, levels 1, 2, 3
ol,, ,	ordered list, unordered list, list item
<a href="page.html">	link to “page.html”
<table>,<tr>,<td>	table, table row, table item
<div>,<span	general containers

HTML Attributes

- HTML elements can have attributes.
- Attributes provide additional information about an element.
- Attributes are always specified in the start tag.
- Attributes come in name/value pairs like: name=“value”



- Sometimes we can find the data we want just by using HTML tags or attributes (e.g, all the <a> tags)
- More often, this isn’t enough: There might be 1000 <a> tags on a page. But maybe we want only the <a> tags *inside* of a <p> tag.
- Enter CSS

17.5.3 CSS

CSS stands for **Cascading Style Sheet**. CSS defines how HTML elements are to be displayed.

HTML came first. But it was only meant to define content, not format it. While HTML contains tags like `<font>` and `<color>`, this is a very inefficient way to develop a website.

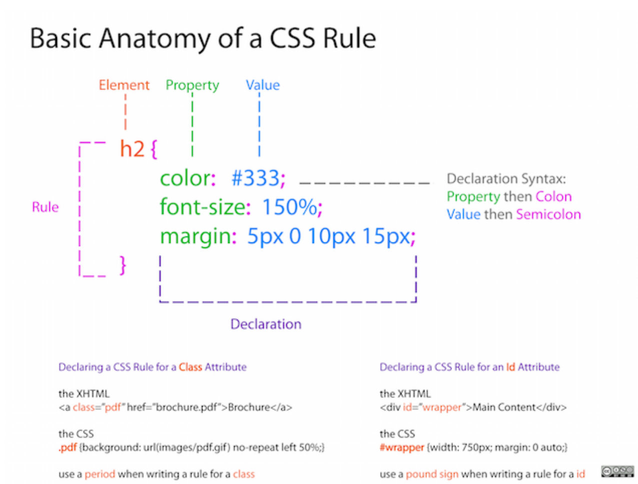
To solve this problem, CSS was created specifically to display content on a webpage. Now, one can change the look of an entire website just by changing one file.

Most web designers litter the HTML markup with tons of `class`es and `id`s to provide "hooks" for their CSS.

You can piggyback on these to jump to the parts of the markup that contain the data you need.

CSS Anatomy

- Selectors
  - Element selector: `p`
  - Class selector: `p class="blue"`
  - I.D. selector: `p id="blue"`
- Declarations
  - Selector: `p`
  - Property: `background-color`
  - Value: `yellow`
- Hooks



17.5.3.1 CSS + HTML

```
<body>
    <table id="content">
        <tr class='name'>
            <td class='firstname'>
                Kurtis
            </td>
            <td class='lastname'>
                McCoy
            </td>
        </tr>
        <tr class='name'>
            <td class='firstname'>
                Leah
            </td>
            <td class='lastname'>
                Guerrero
            </td>
        </tr>
    </table>
</body>
```

Challenge 1

Find the CSS selectors for the following elements in the HTML above.

(Hint: There will be multiple solutions for each)

1. The entire table
2. The row containing "Kurtis McCoy"
3. Just the element containing first names

17.5.4 Finding Elements with Selector Gadget

Selector Gadget is a browser plugin to help you find HTML elements. Install Selector Gadget on your browser by following these instructions.

Once installed, run Selector Gadget and simply click on the type of information you want to select from the webpage. Once this is selected, you can then click the pieces of information you **don't** want to keep. Do this until only the pieces you want to keep remain highlighted, then copy the selector from the bottom pane.

Here's the basic strategy of webscraping:

1. Use Selector Gadget to see how your data is structured
2. Pay attention to HTML tags and CSS selectors

3. Pray that there is some kind of pattern
4. Use R and add-on modules like `RVest` to extract just that data.

Challenge 2

Go to http://rochelleterman.github.io/. Using Selector Gadget,

1. Find the CSS selector capturing all rows in the table.
2. Find the image source URL.
3. Find the HREF attribute of the link.

17.6 Scraping Presidential Statements

To demonstrate webscraping in R, we're going to collect records on presidential statements here: https://www.presidency.ucsb.edu/

Let's say we're interested in how presidents speak about "space exploration". On the website, we punch in this search term, and we get the following 295 results.

Our goal is to scrape these records, and store pertenant information in a dataframe. We will be doing this in two steps:

1. Write a function to scrape each individual record page (these notes).
2. Use this function to loop through all results, and collect all pages (homework).

Load the following packages to get started:

```
library(tidyverse)
library(rvest)
library(stringr)
library(purrr)
library(knitr)
```

17.6.1 Using `RVest` to Read HTML

The package `RVest` allows us to:

1. Collect the HTML source code of a webpage
2. Read the HTML of the page
3. Select and keep certain elements of the page that are of interest

Let's start with step one. We use the `read_html` function to call the results URL and grab the HTML response. Store this result as an object.

```
document1 <- read_html("https://www.presidency.ucsb.edu/documents/special-message-the-congress-re

#Let's take a look at the object we just created
document1
#> {html_document}
#> <html lang="en" dir="ltr" prefix="content: http://purl.org/rss/1.0/modules/content/ dc: http:/
#> [1] <head profile="http://www.w3.org/1999/xhtml/vocab">\n<meta charset=" ...
#> [2] <body class="html not-front not-logged-in one-sidebar sidebar-first  ...
```

This is pretty messy. We need to use `RVest` to make this information more useable.

17.6.2 Find Page Elements

`RVest` has a number of functions to find information on a page. Like other webscraping tools, RVest lets you find elements by their:

1. HTML tags
2. HTML Attributes
3. CSS Selectors

Let's search first for HTML tags.

The function `html_nodes` searches a parsed HTML object to find all the elements with a particular HTML tag, and returns all of those elements.

What does the example below do?

```
html_nodes(document1, "a")
#> {xml_nodeset (78)}
#>  [1] <a href="#main-content" class="element-invisible element-focusable" ...
#>  [2] <a href="https://www.presidency.ucsb.edu/">The American Presidency  ...
#>  [3] <a class="btn btn-default" href="https://www.presidency.ucsb.edu/ab ...
#>  [4] <a class="btn btn-default" href="/advanced-search"><span class="gly ...
#>  [5] <a href="https://www.ucsb.edu/" target="_blank"><img alt="ucsb word ...
#>  [6] <a href="/documents" class="active-trail dropdown-toggle" data-togg ...
#>  [7] <a href="/documents/presidential-documents-archive-guidebook">Guide ...
#>  [8] <a href="/documents/category-attributes">Category Attributes</a>
#>  [9] <a href="/documents/proclamation-9895-national-maritime-day-2019">D ...
#> [10] <a href="/statistics">Statistics</a>
#> [11] <a href="/media" title="">Media Archive</a>
#> [12] <a href="/presidents" title="">Presidents</a>
#> [13] <a href="/analyses" title="">Analyses</a>
#> [14] <a href="https://giving.ucsb.edu/Funds/Give?id=185" title="">Suppor ...
#> [15] <a href="/documents/presidential-documents-archive-guidebook" title ...
#> [16] <a href="/documents" title="" class="active-trail">Categories</a>
#> [17] <a href="/documents/category-attributes" title="">Attributes</a>
```

```
#> [18] <a href="/documents/app-categories/presidential" title="Presidentia ...
#> [19] <a href="/documents/app-categories/spoken-addresses-and-remarks/pre ...
#> [20] <a href="/documents/app-categories/spoken-addresses-and-remarks/pre ...
#> ...
```

That's a lot of results! Many elements on a page will have the same HTML tag. For instance, if you search for everything with the `a` tag, you're likely to get a lot of stuff, much of which you don't want.

In our case, we only want the links corresponding to the speaker Dwight D. Eisenhower.

Using selector gadget, we found out that the CSS selector for document's speaker is `.diet-title a`.

We can then modify our argument in `html_nodes` to look for this more specific CSS selector.

```
html_nodes(document1, ".diet-title a")
#> {xml_nodeset (1)}
#> [1] <a href="/people/president/dwight-d-eisenhower">Dwight D. Eisenhower ...
```

17.6.3 Get Attributes and Text of Elements

Once we identify elements, we want to access information in that element. Oftentimes this means two things:

1) Text
2) Attributes

Getting the text inside an element is pretty straightforward. We can use the `html_text()` command inside of `RVest` to get the text of an element:

```
#Scrape individual document page
document1 <- read_html("https://www.presidency.ucsb.edu/documents/special-message-the-

#identify element with Speaker name
```

```
speaker <- html_nodes(document1, ".diet-title a") %>%
  html_text() #select text of element

speaker
#> [1] "Dwight D. Eisenhower"
```

You can access a tag's attributes using `html_attr`. For example, we often want to get a URL from an `a` (link) element. This is the URL is the link "points" to. It's contained in the attribut `href`:

```
speaker_link <- html_nodes(document1, ".diet-title a") %>%
  html_attr("href")

speaker_link
#> [1] "/people/president/dwight-d-eisenhower"
```

17.6.4 Let's DO this.

Believe it or not, that's all you need to scrape a website. Let's apply these skills to scrape a sample document from the UCSB website – the first item in our search results.

We'll collect the document's date, speaker, title, and full text.

1. Date

```
document1 <- read_html("https://www.presidency.ucsb.edu/documents/special-message-the-congress-re

date <- html_nodes(document1, ".date-display-single") %>%
  html_text() %>% # grab element text
  mdy() #format using lubridate
date
#> [1] "1958-04-02"
```

2. Speaker

```
#Speaker
speaker <- html_nodes(document1, ".diet-title a") %>%
  html_text()
speaker
#> [1] "Dwight D. Eisenhower"
```

3. Title

```
#Title
title <- html_nodes(document1, "h1") %>%
  html_text()
```

```
title
#> [1] "Special Message to the Congress Relative to Space Science and Exploration."
```

4. Text

```
#Text
text <- html_nodes(document1, "div.field-docs-content") %>%
          html_text()

#this is a long document, so let's just display the first 1000 characters
text %>% substr(1, 1000)
#> [1] "\n    To the Congress of the United States:\nRecent developments in long-range
```

17.6.5 Challenge 1: Make a function

Make a function called `scrape_docs` that accepts a URL of an individual document, scrapes the page, and returns a list containing the document's date, speaker, title, and full text.

This involves:

- Requesting the HTML of the webpage using the full URL and RVest.
- Using RVest to locate all elements on the page we want to save.
- Storing each of these items into a list.
- Returning this list.

```
scrape_docs <- function(URL){

  # YOUR CODE HERE

}

# uncomment to test
# scrape_doc("https://www.presidency.ucsb.edu/documents/letter-t-keith-glennan-adminis
```

Chapter 18

Text Analysis

This unit focuses on computational text analysis (or "text-as-data"). We'll explore:

1. **Preprocessing** a corpus for common text analysis.
2. **Sentiment Analysis and Dictionary Methods**, a simple, supervised method for classification.
3. **Distinctive Words**, or word-separating techniques to compare corpora.
4. **Structural Topic Models**, a popular unsupervised method for text exploration and analysis.

These materials are based off a longer, week-long intensive workshop on computational text analysis. If you're interested in text-as-data, I would encourage you to work through these materials on your own: https://github.com/rochelleterman/FSUtext

18.1 Preprocessing

First let's load our required packages.

```
library(tm) # Framework for text mining
library(tidyverse) # Data preparation and pipes $>$
library(ggplot2) # for plotting word frequencies
library(wordcloud) # wordclouds!
```

A **corpus** is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare.

Within each corpus we will have separate articles, stories, volumes, each treated as a separate entity or record. Each unit is called a **document.**

For this unit, we will be using a section of Machiavelli's Prince as our corpus. Since The Prince is a monograph, we have already "chunked" the text, so that each short paragraph or "chunk" is considered a "document."

18.1.1 From Words to Numbers

Corpus Readers

The `tm` package supports a variety of sources and formats. Run the code below to see what it includes

```
getSources()
#> [1] "DataframeSource" "DirSource"       "URISource"       "VectorSource"
#> [5] "XMLSource"       "ZipSource"
getReaders()
#>  [1] "readDataframe"           "readDOC"
#>  [3] "readPDF"                 "readPlain"
#>  [5] "readRCV1"                "readRCV1asPlain"
#>  [7] "readReut21578XML"        "readReut21578XMLasPlain"
#>  [9] "readTagged"              "readXML"
```

Here we'll be reading documents from a csv file. Each row being a document, and columns for text and metadata (information about each document). This is the easiest option if you have metadata.

```
docs.df <-read.csv("data/mach.csv", header=TRUE) #read in CSV file
# docs.df$text <- Encoding("UTF-8")
docs <- Corpus(VectorSource(docs.df$text))
docs
#> <<SimpleCorpus>>
#> Metadata:  corpus specific: 1, document level (indexed): 0
#> Content:  documents: 188
```

Once we have the corpus, we can inspect the documents using inspect()

```
# see the 16th document
inspect(docs[16])
#> <<SimpleCorpus>>
#> Metadata:  corpus specific: 1, document level (indexed): 0
#> Content:  documents: 1
#>
#> [1]  Therefore, since a ruler cannot both practise this virtue of generosity and be
```

Preprocessing functions

Many text analysis applications follow a similar 'recipe' for preprecessing, involving (the order of these steps might differ as per application):

1. Tokenizing the text to unigrams (or bigrams, or trigrams)
2. Converting all characters to lowercase
3. Removing punctuation
4. Removing numbers
5. Removing Stop Words, inclugind custom stop words
6. "Stemming" words, or lemmitization. There are several stemming alogrithms. Porter is the most popular.
7. Creating a Document-Term Matrix

`tm` lets us convert a corpus to a DTM while completing the pre-processing steps in one step.

```
dtm <- DocumentTermMatrix(docs,
           control = list(stopwords = TRUE,
                          tolower = TRUE,
                          removeNumbers = TRUE,
                          removePunctuation = TRUE,
                          stemming=TRUE))
```

Weighting

One common pre-processing step that some applicaitons may call for is applying tf-idf weights. The tf-idf, or term frequency-inverse document frequency, is a weight that ranks the importance of a term in its contextual document corpus. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. In other words, it places importance on terms frequent in the document but rare in the corpus.

```
dtm.weighted <- DocumentTermMatrix(docs,
           control = list(weighting =function(x) weightTfIdf(x, normalize = TRUE),
                          stopwords = TRUE,
                          tolower = TRUE,
                          removeNumbers = TRUE,
                          removePunctuation = TRUE,
                          stemming=TRUE))
```

Compare first 5 rows and 5 columns of the `dtm` and `dtm.weighted`. What do you notice?

```
inspect(dtm[1:5,1:5])
#> <<DocumentTermMatrix (documents: 5, terms: 5)>>
#> Non-/sparse entries: 3/22
#> Sparsity           : 88%
#> Maximal term length: 7
#> Weighting          : term frequency (tf)
#> Sample             :
#>     Terms
#> Docs abandon abil abject abl ablest
#>    1       0    0      0   0      0
#>    2       0    1      0   0      0
#>    3       0    0      0   0      0
#>    4       0    1      0   1      0
#>    5       0    0      0   0      0
inspect(dtm.weighted[1:5,1:5])
#> <<DocumentTermMatrix (documents: 5, terms: 5)>>
#> Non-/sparse entries: 3/22
#> Sparsity           : 88%
#> Maximal term length: 7
#> Weighting          : term frequency - inverse document frequency (normalized) (tf-i
#> Sample             :
#>     Terms
#> Docs abandon   abil abject    abl ablest
#>    1       0 0.0000      0 0.0000      0
#>    2       0 0.0402      0 0.0000      0
#>    3       0 0.0000      0 0.0000      0
#>    4       0 0.0310      0 0.0228      0
#>    5       0 0.0000      0 0.0000      0
```

18.1.2 Exploring the DTM

Dimensions

Let's look at the structure of our DTM. Print the dimensions of the DTM. How many documents do we have? How many terms?

```
# how many documents? how many terms?
dim(dtm)
#> [1]  188 2368
```

Frequencies

We can obtain the term frequencies as a vector by converting the document term matrix into a matrix and using `colSums` to sum the column counts:

```
# how many terms?
freq <- colSums(as.matrix(dtm))
freq[1:5]
#> abandon    abil  abject     abl  ablest
#>       4      35       1      61       1
length(freq)
#> [1] 2368
```

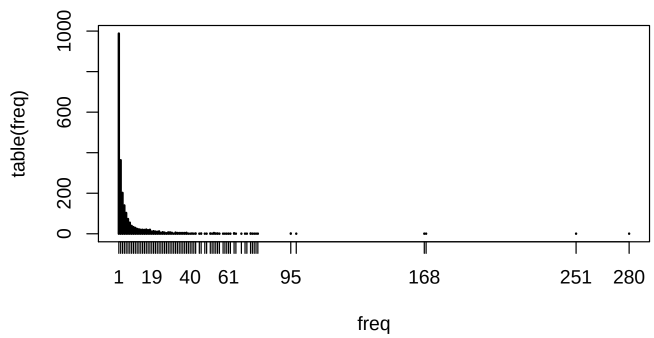
By ordering the frequencies we can list the most frequent terms and the least frequent terms.

```
# order
sorted <- sort(freq, decreasing = T)

# Least frequent terms
head(sorted)
#> ruler  will power   one peopl alway
#>   280   251   169   168    98    95

# most frequent
tail(sorted)
#>  xxiv   xxv  xxvi yield  yoke youth
#>     1     1     1     1     1     1
```

Plotting frequencies

Let's make a plot that shows the frequency of frequencies for the terms. (For example, how many words are used only once? 5 times? 10 times?)

```
# frequency of frenquencies
head(table(freq),15)
#> freq
#>   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
#> 988 363 202 140 103  73  55  39  33  29  24  22  20  20  19
tail(table(freq),15)
#> freq
#>  65  68  70  71  73  74  75  76  77  95  98 168 169 251 280
#>   1   1   1   1   2   1   1   1   1   2   1   1   1   1   1

# plot
plot(table(freq))
```



What does this tell us about the nature of language?

We can reorder columns of DTM to show most frequent terms first:

```
dtm.ordered <- dtm[,order(freq, decreasing = T)]
inspect(dtm.ordered[1:5,1:5])
#> <<DocumentTermMatrix (documents: 5, terms: 5)>>
#> Non-/sparse entries: 10/15
#> Sparsity           : 60%
#> Maximal term length: 5
#> Weighting          : term frequency (tf)
#> Sample             :
#>     Terms
#> Docs one peopl power ruler will
#>    1   0     0     0     1    1
#>    2   3     0     0     1    3
#>    3   0     0     0     0    0
#>    4   0     0     0     1    1
#>    5   3     0     0     1    1
```

Exploring common words

The TM package has lots of useful functions to help you explore common words and associations:

```
# Have a look at common words
findFreqTerms(dtm, lowfreq=50) # words that appear at least 100 times
#>  [1] "abl"      "act"      "alway"    "armi"     "becom"    "can"
#>  [7] "consid"   "either"   "forc"     "great"    "king"     "maintain"
#> [13] "make"     "man"      "mani"     "men"      "much"     "must"
#> [19] "never"    "new"      "one"      "order"    "other"    "peopl"
#> [25] "power"    "reason"   "ruler"    "sinc"     "state"    "subject"
#> [31] "time"     "troop"    "use"      "want"     "way"      "well"
#> [37] "will"
```

```
# Which words correlate with "war"?
findAssocs(dtm, "war", 0.3)
#> $war
#>     wage    fight  antioch     argu     brew    induc      lip    maxim
#>     0.73     0.52     0.45     0.45     0.45     0.45     0.45     0.45
#>  relianc     sage    trifl  postpon     mere     evil    avoid     flee
#>     0.45     0.45     0.45     0.41     0.35     0.34     0.32     0.32
#>   occupi     glad glorious    heard     hunt  ineffect     knew   produc
#>     0.32     0.30     0.30     0.30     0.30     0.30     0.30     0.30
#> temporis
#>     0.30
```

We can even make wordclouds showing the most commons terms:

```
# wordcoulds!
set.seed(123)
wordcloud(names(sorted), sorted, max.words=100, colors=brewer.pal(6,"Dark2"))
#> Warning in wordcloud(names(sorted), sorted, max.words = 100, colors =
#> brewer.pal(6, : ruler could not be fit on page. It will not be plotted.
#> Warning in wordcloud(names(sorted), sorted, max.words = 100, colors =
#> brewer.pal(6, : power could not be fit on page. It will not be plotted.
```

Remove sparse terms

Somtimes we want to remove sparse terms and thus inrease efficency. Look up the help file for the function `removeSparseTerms`. Using this function, create an objected called `dtm.s` that contains only terms with <.9 sparsity (meaning they appear in more than 10% of documents).

```
dtm.s <- removeSparseTerms(dtm,.9)
dtm # 2365 terms
#> <<DocumentTermMatrix (documents: 188, terms: 2368)>>
#> Non-/sparse entries: 11754/433430
#> Sparsity           : 97%
#> Maximal term length: 15
#> Weighting          : term frequency (tf)
dtm.s # 135 terms
#> <<DocumentTermMatrix (documents: 188, terms: 136)>>
#> Non-/sparse entries: 4353/21215
#> Sparsity           : 83%
#> Maximal term length: 12
#> Weighting          : term frequency (tf)
```

18.1.3 Exporting the DTM

We can convert a DTM to a matrix or data.frame in order to write to a csv, add meta data, etc.

First create an object that converts the `dtm` to a dataframe (we first have to convert to matrix, and then to dataframe)

```
# coerce into dataframe
dtm <- as.data.frame(as.matrix(dtm))
names(dtm)[1:10]  # names of documents
#>  [1] "abandon"   "abil"      "abject"    "abl"       "ablest"
#>  [6] "abovement" "abovenam"  "absolut"   "absorb"    "accept"
```

Now add a column called `doc_section`. For the first 100 rows, the value of this column should be "Section 1". For documents 101-188, the section should be "Section 2".

```
# add fake column for section
dtm$doc_section <- "NA"
dtm$doc_section[1:100] <- "Section 1"
dtm$doc_section[101:188] <- "Section 2"
dtm$doc_section <- as.factor(dtm$doc_section)

# check to see if they're the same number of documents per author
summary(dtm$doc_section)
```

```
#> Section 1 Section 2
#>       100        88
```

We can now export the dataframe as a csv.

```
write.csv(dtm, "dtm.csv", row.names = F)
```

18.1.4 Challenges

Using the one of the datasetes in the `text-data` repo, create a document term matrix and create a wordcloud of the most common terms.

```
# download text-data
library("usethis")
use_course("plsc-31101/text-data")

# check out what you just downloaded.
# and unzip a dataset.

# YOUR CODE HERE
```

18.2 Sentiment Analysis and Dictionary Methods

To demonstrate sentiment analysis, we're going to explore lyrics from Taylor Swift songs.

Road the code below to get started.

```
require(tm)
require(tidytext)
require(tidyverse)
require(stringr)
require(textdata)

ts <- read.csv("data/taylor_swift.csv")
getwd()
#> [1] "/Users/rterman/Desktop/course"
```

18.2.1 Preprocessing and Setup

First we must preprocess the corpus. Create a document-term matrix from the `lyrics` column of the `ts` data frame. Complete the following preprocessing steps: - convert to lower - remove stop words - remove numbers - remove punctuation.

Think: Why is stemming inappropriate for this application?

```
# preprocess and create DTM
docs <- Corpus(VectorSource(ts$lyrics))

dtm <- DocumentTermMatrix(docs,
           control = list(tolower = TRUE,
                          removeNumbers = TRUE,
                          removePunctuation = TRUE,
                          stopwords = TRUE
                         ))

dtm <- as.data.frame(as.matrix(dtm))
```

Sentiment dictionaries

We're going to use sentiment dictionaries from the `tidytext` package. Using the `get_sentiments` function, load the "bing" dictionary and store it in an object called `sent`.

```
sent <- get_sentiments("bing")
head(sent)
#> # A tibble: 6 x 2
#>   word        sentiment
#>   <chr>       <chr>
#> 1 2-faces     negative
#> 2 abnormal    negative
#> 3 abolish     negative
#> 4 abominable negative
#> 5 abominably negative
#> 6 abominate  negative
```

We'll now add a column to `sent` called `score`. This column should hold a "1" for positive words and "-1" for negative words.

```
sent$score <- ifelse(sent$sentiment=="positive", 1, -1)
```

18.2.2 Scoring the songs

We're now ready to score each song.

(**NB**: There are probably many ways to program a script that performs this task. If you can think of a more elegant way, go for it!)

First, we'll create a dataframe that holds all the words in our dtm along with their sentiment score.

```
# get all the words in our dtm and put it in a dataframe
words = data.frame(word = colnames(dtm))
head(words)
#>        word
#> 1      back
#> 2 backroads
#> 3       bed
#> 4   believe
#> 5   beneath
#> 6    beside

# get their sentiment scores
words <- merge(words, sent, all.x = T)
head(words)
#>       word sentiment score
#> 1  abigail      <NA>    NA
#> 2   absent      <NA>    NA
#> 3   absurd  negative    -1
#> 4  accused      <NA>    NA
#> 5      ace      <NA>    NA
#> 6 achilles      <NA>    NA

# replace NAs with 0s
words$score[is.na(words$score)] <- 0
head(words)
#>       word sentiment score
#> 1  abigail      <NA>     0
#> 2   absent      <NA>     0
#> 3   absurd  negative    -1
#> 4  accused      <NA>     0
#> 5      ace      <NA>     0
#> 6 achilles      <NA>     0
```

We can now use matrix algebra (!!) to multiply our dtm by the scoring vector. This will return to us a score for each document (i.e., song).

```
# calculate documents scores with matrix algebra!
scores <- as.matrix(dtm) %*% words$score

# put it in the original documents data frame
ts$sentiment <- scores
```

Which song is happiest? Go listen to the song and see if you agree.

18.2.3 Challenges

Challenge 1

Using the code we wrote above, make a function that accepts 1) a vector of texts, and 2) a sentiment dictionary (i.e. a data frame with words and scores), and returns a vector of sentiment scores for each text.

```
sentiment_score <- function(texts, sent_dict){

  # YOUR CODE HERE

  return(scores)
}

# uncomment to test it out!
# sentiment_score(ts$lyrics, sent)
```

Challenge 2

Using the function you wrote above, find out what the most and least positive Taylor Swift album is.

```
# concatenate to make albums
albums <- ts %>%
  group_by(album) %>%
  summarise(lyrics = str_c(lyrics, collapse = ";"))

# first load the dictionary
afinn <- get_sentiments("afinn")

# then run the function
sentiment_score(albums$lyrics, afinn)

# add to original df
albums$sent <- sentiment_score(albums$lyrics, afinn)
```

18.3 Distinctive Words

This lesson finds distinctive words in the speeches of Obama and Trump.

Run the following code to:

1. Import the corpus
2. Create a DTM

```r
require(tm)
require(matrixStats) # for statistics

# import corpus
docs <- Corpus(DirSource("Data/trump_obama"))

# preprocess and create DTM
dtm <- DocumentTermMatrix(docs,
           control = list(tolower = TRUE,
                          removePunctuation = TRUE,
                          removeNumbers = TRUE,
                          stopwords = TRUE,
                          stemming=TRUE))

# print the dimensions of the DTM
dim(dtm)
#> [1]   11 4094

# take a quick look
inspect(dtm[,100:104])
#> <<DocumentTermMatrix (documents: 11, terms: 5)>>
#> Non-/sparse entries: 14/41
#> Sparsity           : 75%
#> Maximal term length: 11
#> Weighting          : term frequency (tf)
#> Sample             :
#>                 Terms
#> Docs              alien align alik aliv allamerican
#>   Obama_2009.txt     0     0    1    0           0
#>   Obama_2010.txt     0     0    1    1           0
#>   Obama_2011.txt     0     1    0    0           0
#>   Obama_2012.txt     0     0    0    1           0
#>   Obama_2013.txt     0     0    0    0           0
#>   Obama_2014.txt     0     0    0    1           0
#>   Obama_2015.txt     1     0    1    0           0
#>   Trump_2017.txt     0     1    0    0           0
#>   Trump_2018.txt     1     0    0    0           1
#>   Trump_2019.txt     3     0    1    1           0
```

Oftentimes scholars will want to compare different corpora by finding the words (or features) distinctive to each corpora. But finding distinctive words requires a decision about what "distinctive" means. As we will see, there are a variety of definitions that we might use.

18.3.1 Unique usage

The most obvious definition of distinctive is "exclusive". That is, distinctive words are those are found exclusively in texts associated with a single speaker (or group). For example, if Trump uses the word "access" and Obama never does, we should count "access" as distinctive.

Finding words that are exclusive to a group is a simple exercise. All we have to do is sum the usage of each word use across all texts for each speaker, and then look for cases where the sum is zero for one speaker.

```
# turn DTM into dataframe
dtm.m <- as.data.frame(as.matrix(dtm))
dtm.m$that <- NULL # fix weird error with stopwords.

# Subset into 2 dtms for each speaker
obama <- dtm.m[1:8,]
trump <- dtm.m[9:11,]

# Sum word usage counts across all texts
obama <- colSums(obama)
trump <- colSums(trump)

# Put those sums back into a dataframe
df <- data.frame(rbind(obama, trump))
df[ ,1:5]
#>       abandon abess abid abil object
#> obama       2     1    1    7      0
#> trump       1     0    0    1      1

# Get words where one speaker's usage is 0
solelyobama <- unlist(df[1, trump==0])
solelyobama <- solelyobama[order(solelyobama, decreasing = T)] # order them by frequen
head(solelyobama, 10) # get top 10 words for obama
#>      dont technolog      bank     innov    doesnt   teacher      loan 
#>        68        31        30        30        29        26        22 
#>      wont     debat    climat 
#>        22        21        19 

solelytrump <- unlist(df[2, obama==0])
solelytrump <- solelytrump[order(solelytrump, decreasing = T)] # order them by frequen
head(solelytrump, 10) # get top 10 words for trump
#>      isi    agent america.  audienc    megan      it. obamacar     alic 
#>        9        8        8        8        8        7        7        6 
#>   beauti    elvin 
#>        6        6 
```

As we can see, these words tend not to be terribly interesting or informative. So we will remove them from our corpus in order to focus on identifying distinctive words that appear in texts associated with every speaker.

```
# subset df with non-zero entries
df <- df[,trump>0 & obama>0]

# how many words are we left with?
ncol(df)
#> [1] 1525
df[,1:5]
#>       abandon abil abl abort abraham
#> obama       2    7  15     1       1
#> trump       1    1   9     1       1
```

18.3.2 Differences in frequences

Another basic approach to identifying distinctive words is to compare the frequencies at which speakers use a word. If one speaker uses a word often across his or her oeuvre and another barely uses the word at all, the difference in their respective frequencies will be large. We can calculate this quantity the following way:

```
# take the differences in frequences
diffFreq <- obama - trump

# sort the words
diffFreq <- sort(diffFreq, decreasing = T)

# the top obama words
head(diffFreq, 10)
#>     will     year      job     work     make      can american     that
#>      306      217      214      186      177      172      165      160
#>  america      new
#>      155      150

# the top trump words
tail(diffFreq, 10)
#>  illeg immigr    isi    usa   hero   ryan border  great  thank   drug
#>     -9     -9     -9     -9    -11    -11    -13    -13    -19    -22
```

18.3.3 Differences in averages

This is a good start. But what if one speaker uses more words *overall*? Instead of using raw frenquncies, a better approach would look at the average *rate* at which speakers use various words.

We can calculate this quantity the following way:

1. Normalize the DTM from counts to proportions
2. Take the difference between one speaker's proportion of a word and another's proportion of the same word.
3. Find the words with the highest absolute difference.

```
# normalize into proportions
rowTotals <- rowSums(df) #create vector with row totals, i.e. total number of words pe
head(rowTotals) # notice that one speaker uses more words than the other
#> obama trump
#> 23021  7432

# change frequencies to proportions
df <- df/rowTotals #change frequencies to proportions
df[,1:5]
#>        abandon     abil      abl    abort  abraham
#> obama 8.69e-05 0.000304 0.000652 4.34e-05 4.34e-05
#> trump 1.35e-04 0.000135 0.001211 1.35e-04 1.35e-04

# get difference in proportions
means.obama <- df[1,]
means.trump <- df[2,]
score <- unlist(means.obama - means.trump)

# find words with highest difference
score <- sort(score, decreasing = T)
head(score,10) # top obama words
#>     job    make    busi     let    need    work    help economi  energi
#> 0.00620 0.00541 0.00473 0.00426 0.00419 0.00407 0.00388 0.00378 0.00363
#>     can
#> 0.00346
tail(score,10) # top trump words
#>   border  tonight   immigr     unit    state     drug     must    great
#> -0.00284 -0.00293 -0.00322 -0.00322 -0.00322 -0.00342 -0.00354 -0.00476
#>    thank american
#> -0.00483 -0.00650
```

This is a start. The problem with this measure is that it tends to highlight differences in very frequent words. For example, this method gives greater attention to a word that occurs 30 times per 1,000 words in Obama and 25 times per 1,000 in Trump than it does to a word that occurs 5 times per 1,000 words in Obama and 0.1 times per 1,000 words in Trump This does not seem right. It seems important to recognize cases when one speaker uses a word frequently and another speaker barely uses it.

As this initial attempt suggests, identifying distinctive words will be a balancing act. When comparing two groups of texts, differences in the rates of frequent words will tend to be large relative to differences in the rates of rarer words. Human language is variable; some words occur more frequently than others regardless of who is writing. We need to find a way of adjusting our definition of distinctive in light of this.

18.3.4 Difference in averages, adjustment

One adjustment that is easy to make is to divide the difference in speakers' average rates by the average rate across all speakers. Since dividing a quantity by a large number will make that quantity smaller, our new distinctiveness score will tend to be lower for words that occur frequently. While this is merely a heuristic, it does move us in the right direction.

```
# get the average rate of all words across all speakers
means.all <- colMeans(df)

# now divide the difference in speakers' rates by the average rate across all speakers
score <- unlist((means.obama - means.trump) / means.all)
score <- sort(score, decreasing = T)
head(score,10) # top obama words
#>    student       cant       idea      money        oil     higher
#>       1.78       1.77       1.70       1.67       1.67       1.66
#>       earn leadership   research    respons
#>       1.60       1.60       1.59       1.58
tail(score,10) # top trump words
#>     drug    grace    death    heart   pillar southern  terribl   unfair
#>    -1.77    -1.80    -1.82    -1.82    -1.84    -1.84    -1.84    -1.84
#>     gang     ryan
#>    -1.87    -1.90
```

18.4 Structural Topic Models

This unit gives a brief overview of the `stm` (structural topic model) package. Please read the vignette for more detail.

Structural topic model is a way to estimate a topic model that includes document-level meta-data. One can then see how topical prevalence changes according to that meta-data.

```
library(stm)
```

The data we'll be using for this unit consists of all articles about women published in the New York Times and Washington Post, 1980-2014. You worked with a subset of this data in your last homework.

Load the dataset. Notice that we have the text of the articles, along with some metadata.

```
# Load Data
women <- read.csv('data/women-full.csv')
names(women)
#>  [1] "BYLINE"              "TEXT.NO.NOUN"        "PUBLICATION"
#>  [4] "TITLE"               "COUNTRY"             "COUNTRY_FINAL"
#>  [7] "YEAR"                "UID"                 "COUNTRY_NR"
#> [10] "entities"            "LENGTH"              "COUNTRY_TOP_PERCENT"
#> [13] "COUNTRY_CODE"        "TEXT"                "DATE"
#> [16] "COUNTRY_MAJOR"       "TYPE"                "REGION"
#> [19] "SUBJECT"
```

18.4.1 Preprocessing

STM has its own unique preprocessing functions and procedure, which I've coded below. Notice that we're going to use the `TEXT.NO.NOUN` column, which contains all the text of the articles without proper nouns (which I removed earlier).

```
# Pre-process
temp<-textProcessor(documents = women$TEXT.NO.NOUN, metadata = women)
#> Building corpus...
#> Converting to Lower Case...
#> Removing punctuation...
#> Removing stopwords...
#> Removing numbers...
#> Stemming...
#> Creating Output...
meta<-temp$meta
vocab<-temp$vocab
docs<-temp$documents

# prep documents in correct format
out <- prepDocuments(docs, vocab, meta)
#> Removing 19460 of 39403 terms (19460 of 1087166 tokens) due to frequency
#> Your corpus now has 4531 documents, 19943 terms and 1067706 tokens.
docs<-out$documents
vocab<-out$vocab
meta <-out$meta
```

Challenge 1

Read the help file for the `prepDocuments` function. Alter the code above (in 2.1) to keep only words that appear in at least 10 documents.

```
# YOUR CODE HERE
```

18.4.2 Estimate Model

We're now going to estimate a topic model with 15 topics by regressing topical prevalence on region and year covariates.

Running full model takes a **long** time to finish. For that reason, we're going to add an argument `max.em.its` which sets the number of iterations. By keeping it low (15) we'll see a rough estimate of the topics. You can always go back and estimate the model to convergence.

```
model <- stm(docs, vocab, 15, prevalence = ~ REGION + s(YEAR), data = meta, seed = 15, max.em.its
#> Beginning Spectral Initialization
#>   Calculating the gram matrix...
#>   Using only 10000 most frequent terms during initialization...
#>   Finding anchor words...
#>      ...............
#>   Recovering initialization...
#>      ..............................................................................
#> Initialization complete.
#> ..................................................................................
#> Completed E-Step (7 seconds).
#> Completed M-Step.
#> Completing Iteration 1 (approx. per word bound = -7.882)
#> ..................................................................................
#> Completed E-Step (4 seconds).
#> Completed M-Step.
#> Completing Iteration 2 (approx. per word bound = -7.605, relative change = 3.519e-02)
#> ..................................................................................
#> Completed E-Step (4 seconds).
#> Completed M-Step.
#> Completing Iteration 3 (approx. per word bound = -7.573, relative change = 4.218e-03)
#> ..................................................................................
#> Completed E-Step (3 seconds).
#> Completed M-Step.
#> Completing Iteration 4 (approx. per word bound = -7.559, relative change = 1.843e-03)
#> ..................................................................................
#> Completed E-Step (3 seconds).
#> Completed M-Step.
#> Completing Iteration 5 (approx. per word bound = -7.550, relative change = 1.162e-03)
```

```
#> Topic 1: women, show, one, like, design
#>  Topic 2: said, women, polic, report, year
#>  Topic 3: women, team, said, game, world
#>  Topic 4: year, book, first, one, mrs
#>  Topic 5: women, said, percent, femal, will
#>  Topic 6: said, women, one, peopl, woman
#>  Topic 7: women, work, said, year, men
#>  Topic 8: women, sexual, sex, rape, men
#>  Topic 9: women, said, right, law, govern
#>  Topic 10: said, one, famili, women, peopl
#>  Topic 11: women, film, one, said, woman
#>  Topic 12: said, polit, elect, parti, govern
#>  Topic 13: women, said, cancer, studi, health
#>  Topic 14: women, confer, said, deleg, will
#>  Topic 15: said, women, girl, rape, practic
#> ..........................................................................
#> Completed E-Step (3 seconds).
#> Completed M-Step.
#> Completing Iteration 6 (approx. per word bound = -7.544, relative change = 8.200e-0
#> ..........................................................................
#> Completed E-Step (3 seconds).
#> Completed M-Step.
#> Completing Iteration 7 (approx. per word bound = -7.539, relative change = 5.961e-0
#> ..........................................................................
#> Completed E-Step (3 seconds).
#> Completed M-Step.
#> Completing Iteration 8 (approx. per word bound = -7.536, relative change = 4.491e-0
#> ..........................................................................
#> Completed E-Step (3 seconds).
#> Completed M-Step.
#> Completing Iteration 9 (approx. per word bound = -7.533, relative change = 3.481e-0
#> ..........................................................................
#> Completed E-Step (3 seconds).
#> Completed M-Step.
#> Completing Iteration 10 (approx. per word bound = -7.531, relative change = 2.759e-
#> Topic 1: show, design, women, fashion, art
#>  Topic 2: said, women, polic, report, offici
#>  Topic 3: women, team, said, game, world
#>  Topic 4: book, year, life, first, work
#>  Topic 5: women, said, femal, percent, will
#>  Topic 6: said, one, protest, peopl, site
#>  Topic 7: women, work, said, year, percent
#>  Topic 8: women, sexual, rape, sex, men
#>  Topic 9: women, said, right, law, govern
#>  Topic 10: said, one, famili, peopl, day
```

```
#>  Topic 11: women, film, one, like, woman
#>  Topic 12: polit, said, elect, parti, govern
#>  Topic 13: women, said, abort, cancer, health
#>  Topic 14: women, confer, said, will, world
#>  Topic 15: said, women, girl, rape, case
#> ....................................................................................................
#> Completed E-Step (3 seconds).
#> Completed M-Step.
#> Completing Iteration 11 (approx. per word bound = -7.530, relative change = 2.214e-04)
#> ....................................................................................................
#> Completed E-Step (3 seconds).
#> Completed M-Step.
#> Completing Iteration 12 (approx. per word bound = -7.528, relative change = 1.804e-04)
#> ....................................................................................................
#> Completed E-Step (3 seconds).
#> Completed M-Step.
#> Completing Iteration 13 (approx. per word bound = -7.527, relative change = 1.496e-04)
#> ....................................................................................................
#> Completed E-Step (3 seconds).
#> Completed M-Step.
#> Completing Iteration 14 (approx. per word bound = -7.526, relative change = 1.265e-04)
#> ....................................................................................................
#> Completed E-Step (3 seconds).
#> Completed M-Step.
#> Model Terminated Before Convergence Reached
```

Let's see what our model came up with! The following tools can be used to evaluate the model.

- `labelTopics` gives the top words for each topic.
- `findThoughts` gives the top documents for each topic (the documents with the highest proportion of each topic)

```
# Top Words
labelTopics(model)
#> Topic 1 Top Words:
#>       Highest Prob: show, design, fashion, women, art, one, like
#>       FREX: coutur, fashion, museum, sculptur, ready--wear, jacket, galleri
#>       Lift: ---inch, -ankl, alexandr, armatur, armhol, art-fair, avant
#>       Score: coutur, art, artist, fashion, museum, exhibit, cloth
#> Topic 2 Top Words:
#>       Highest Prob: said, polic, women, kill, report, offici, govern
#>       FREX: polic, suicid, kill, attack, investig, suspect, arrest
#>       Lift: abducte, charanjit, humanity-soak, male-control, sunil, kalpana, ciudad
#>       Score: polic, rape, kill, said, arrest, attack, investig
#> Topic 3 Top Words:
```

```
#>       Highest Prob: women, team, game, said, world, play, olymp
#>       FREX: tournament, championship, olymp, soccer, player, game, medal
#>       Lift: -america, -foot--inch, -hole, -kilomet, -rank, -round, -trump
#>       Score: olymp, championship, tournament, team, player, game, medal
#> Topic 4 Top Words:
#>       Highest Prob: book, year, life, first, write, novel, work
#>       FREX: novel, literari, fiction, book, memoir, novelist, poet
#>       Lift: buster, calla, goncourt, identical-twin, italian-american, kilcher, klo
#>       Score: novel, book, fiction, literari, poet, writer, write
#> Topic 5 Top Words:
#>       Highest Prob: women, said, femal, percent, militari, will, compani
#>       FREX: combat, board, quota, militari, bank, corpor, infantri
#>       Lift: -combat, cpr, gender-divers, nonexecut, outfitt, r-calif, r-ni
#>       Score: women, militari, infantri, combat, percent, quota, femal
#> Topic 6 Top Words:
#>       Highest Prob: protest, said, one, site, peopl, young, video
#>       FREX: orthodox, internet, web, video, rabbi, prayer, site
#>       Lift: balaclava, grrrl, tehrik-, braveheart, drawbridg, gravesit, guerrilla-s
#>       Score: protest, site, orthodox, video, jewish, rabbi, xxxfx
#> Topic 7 Top Words:
#>       Highest Prob: women, work, said, year, percent, men, ese
#>       FREX: ese, factori, employ, incom, worker, job, market
#>       Lift: flextim, management-track, nec, nontransfer, rabenmutt, chiho, fumiko
#>       Score: ese, percent, compani, work, job, women, factori
#> Topic 8 Top Words:
#>       Highest Prob: women, sexual, sex, rape, men, violenc, said
#>       FREX: harass, sexual, sex, assault, brothel, violenc, behavior
#>       Lift: offenc, tarun, chaud, much-lov, newt, tiresom, sex-rel
#>       Score: rape, sexual, harass, violenc, sex, assault, brothel
#> Topic 9 Top Words:
#>       Highest Prob: women, said, right, law, islam, govern, religi
#>       FREX: islam, religi, veil, constitut, saudi, secular, cleric
#>       Lift: afghan-styl, anglo-, archdeacon, bien-aim, episcopaci, fez, government-
#>       Score: islam, law, women, right, religi, ordin, saudi
#> Topic 10 Top Words:
#>       Highest Prob: said, one, famili, peopl, day, like, home
#>       FREX: villag, room, smile, son, couldnt, recal, sit
#>       Lift: charpoy, jet-black, mitra, schermerhorn, single-famili, tyson, uja-fede
#>       Score: villag, husband, fistula, famili, school, girl, said
#> Topic 11 Top Words:
#>       Highest Prob: women, film, one, like, woman, say, play
#>       FREX: film, theater, movi, charact, actress, documentari, audienc
#>       Lift: clive, fine-tun, kaffir, nushus, shrew, nushu, cadel
#>       Score: film, theater, movi, nushu, play, orchestra, femin
#> Topic 12 Top Words:
```

```
#>       Highest Prob: polit, elect, parti, minist, presid, govern, said
#>       FREX: voter, elect, parti, prime, candid, vote, cabinet
#>       Lift: ernesto, pinbal, influence-peddl, information-servic, kakuei, left--cent, marxist-
#>       Score: elect, parti, vote, minist, voter, polit, candid
#> Topic 13 Top Words:
#>       Highest Prob: women, said, abort, cancer, health, studi, breast
#>       FREX: implant, cancer, breast, pill, virus, patient, estrogen
#>       Lift: acet, adren, ambulatori, analges, anastrozol, antioxid, ashkenazi
#>       Score: cancer, abort, breast, pill, implant, health, virus
#> Topic 14 Top Words:
#>       Highest Prob: women, said, confer, will, world, organ, right
#>       FREX: deleg, confer, forum, page, peac, nongovernment, ambassador
#>       Lift: -glass, barack, brooklyn-born, expansion, foreclosur, guarantor, holden
#>       Score: deleg, confer, forum, page, palestinian, peac, mrs
#> Topic 15 Top Words:
#>       Highest Prob: said, women, rape, court, case, girl, practic
#>       FREX: mutil, genit, circumcis, asylum, sentenc, judg, tribun
#>       Lift: labia, layli, minora, multifaith, paraleg, salim, strip-search
#>       Score: rape, genit, circumcis, mutil, court, sentenc, prosecutor

# Example Docs
findThoughts(model, texts = meta$TITLE, n=2,topics = 1:15)
#> Warning in findThoughts(model, texts = meta$TITLE, n = 2, topics =
#> 1:15): texts are of type 'factor.' Converting to character vectors. Use
#> 'as.character' to avoid this warning in the future.
#>
#>  Topic 1:
#>       KENZO'S CAREFREE STYLES AT AN OFFBEAT SHOWING
#>      A MODERN LOOK, A CLASSIC TOUCH FROM SAINT LAURENT
#>  Topic 2:
#>       Assailants Kill 4 Iraqi Women Working for U.S.; Gunmen Follow Van Carrying Laundry Emplo
#>      WORLD IN BRIEF
#>  Topic 3:
#>       AMERICANS LEAD EAST GERMANS IN TRACK
#>      Russians Chart a New Path
#>  Topic 4:
#>       BEST SELLERS: September 6, 1998
#>      BEST SELLERS: September 13, 1998
#>  Topic 5:
#>       In Britain, a Big Push for More Women to Serve on Corporate Boards
#>      Poll: Allow women in combat units
#>  Topic 6:
#>       Neda's Legacy; A woman's death moves Iranian protesters.
#>      Jewish Feminists Prompt Protests at Wailing Wall
#>  Topic 7:
```

```
#>       China Scrambles for Stability as Its Workers Age
#>      A high price for a paycheck; Caught between the demands of the corporate workp
#>  Topic 8:
#>       Confronting Rape in India, and Around the World
#>      Sexual Harassment Prosecutions Get Short Shrift in India, Lawyer Says
#>  Topic 9:
#>       English Church Advances Bid For Women As Bishops
#>      Egypt Passes Law On Women's Rights; Polygamy Still Allowed for Men
#>  Topic 10:
#>       An Old Cinema in Pakistan Has New Life After Quake
#>      Maria Duran's Endless Wait
#>  Topic 11:
#>       For France, An All-Purpose Heartthrob
#>      Film: Brazilian 'Vera'
#>  Topic 12:
#>       The Widow Of Ex-Leader Wins Race In Panama
#>      Cabinet Defeated in Iceland as Feminists Gain
#>  Topic 13:
#>       SECTION: HEALTH; Pg. T18
#>      Dense Breasts May Need Sonograms to Detect Cancer
#>  Topic 14:
#>       DISPUTES ON KEY ISSUES STALL KENYA PARLEY
#>      'CHAOTIC' CONDITIONS FEARED AT U.N.'S PARLEY ON WOMEN
#>  Topic 15:
#>       Woman Fleeing Tribal Rite Gains Asylum; Genital Mutilation Is Ruled Persecuti
#>      Refugee From Mutilation
```

18.4.2.1 Challenge 2

Estimate other models using 5 and 40 topics, respectively. Look at the top words for each topic. How do the topics vary when you change the number of topics?

Now look at your neighbor's model. Did you get the same results? Why or why not?

```
# YOUR CODE HERE
```

18.4.3 Interprete Model

Let's all load a fully-estimated model that I ran before class.

```
# load the already-estimated model.
load("data/stm.RData")
```

Challenge 3

Using the functions `labelTopics` and `findThoughts`, hand label the 15 topics. Hold these labels as a character vector called `labels`

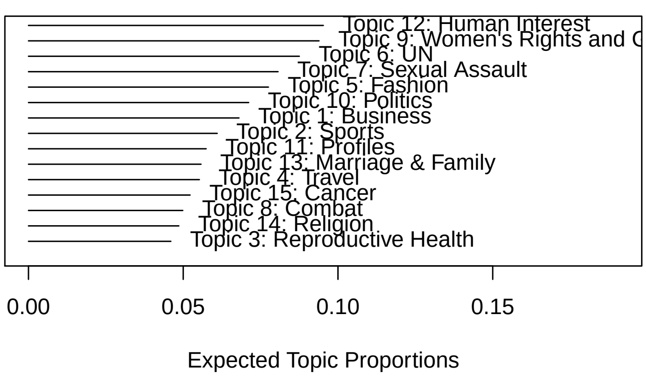
```
# Store your hand labels below.
labels = c()
```

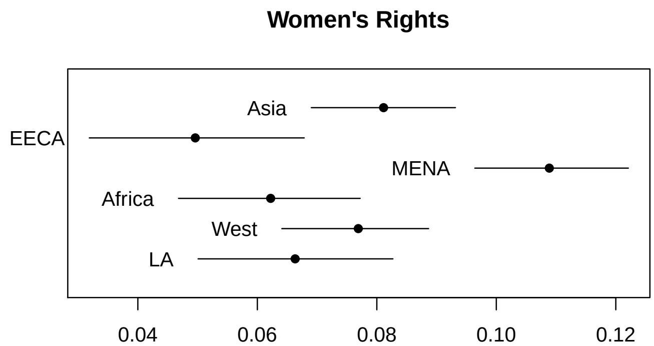
Now look at your neighbor's labels. Did you get the same results? Why or why not?

18.4.4 Analyze topics

We're now going to see how the topics compare in terms of their prevalence across region. What do you notice about the distribution of topic 9?

```
# Corpus Summary
plot.STM(model, type="summary", custom.labels = labels, main="")

# Estimate Covariate Effects
prep <- estimateEffect(1:15 ~ REGION + s(YEAR), model, meta = meta, uncertainty = "Global", docum

# plot topic 9 over region
regions = c("Asia", "EECA", "MENA", "Africa", "West", "LA")
plot.estimateEffect(prep, "REGION", method = "pointestimate", topics = 9, printlegend = TRUE, lab
```




Women's Rights
Asia
EECA
MENA
Africa
West
LA
0.04
0.06
0.08
0.10
0.12

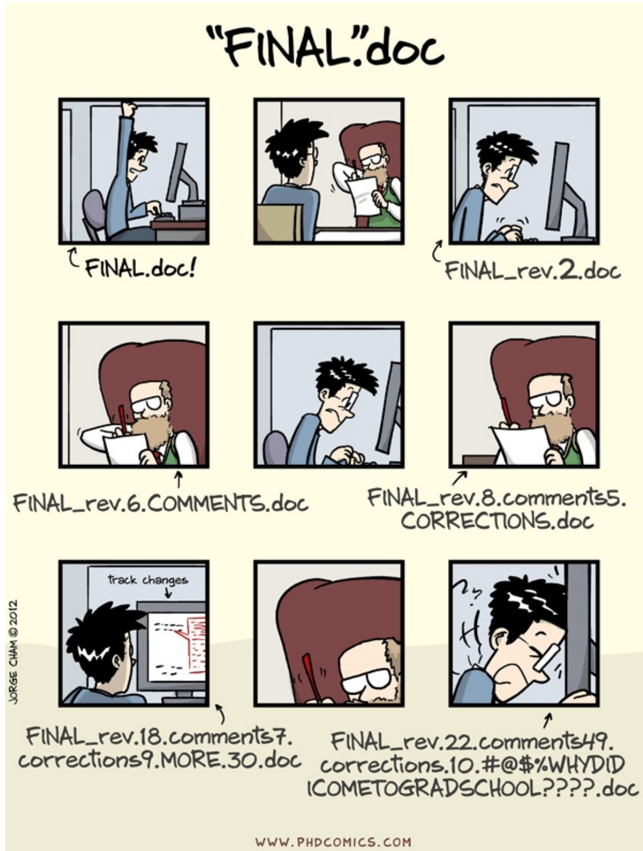
Chapter 19

Git and Github

19.1 Learning Objectives

- Explain which initialization and configuration steps are required once per machine, and which are required once per repository.
- Go through the modify-add-commit cycle for single and multiple files and explain where information is stored at each stage.
- Configure Git to ignore specific files, and explain why it is sometimes useful to do so.
- Explain collaboration options on GitHub
- Go through the fork & pull workflow

We'll start by exploring how version control can be used to keep track of what one person did and when. Even if you aren't collaborating with other people, version control is much better for this than this:



Git is powerful and complicated. We can do a full day workshop on git alone. But it is also quite possible to harness it's powers by cycing through three commands: `add`, `commit`, `push`. So even if you don't understand what's going on underneath the hood, knowing just these three commands can get you very far.

19.2 Starting with Git

The first time we use Git on a new machine, we need to configure a few things.

Here's how Dracula sets up his new laptop:

```
$ git config --global user.name "Vlad Dracula"
$ git config --global user.email "vlad@tran.sylvan.ia"
```

(Please use your own name and email address instead of Dracula's, and please use the same email as you used to make your GitHub account.)

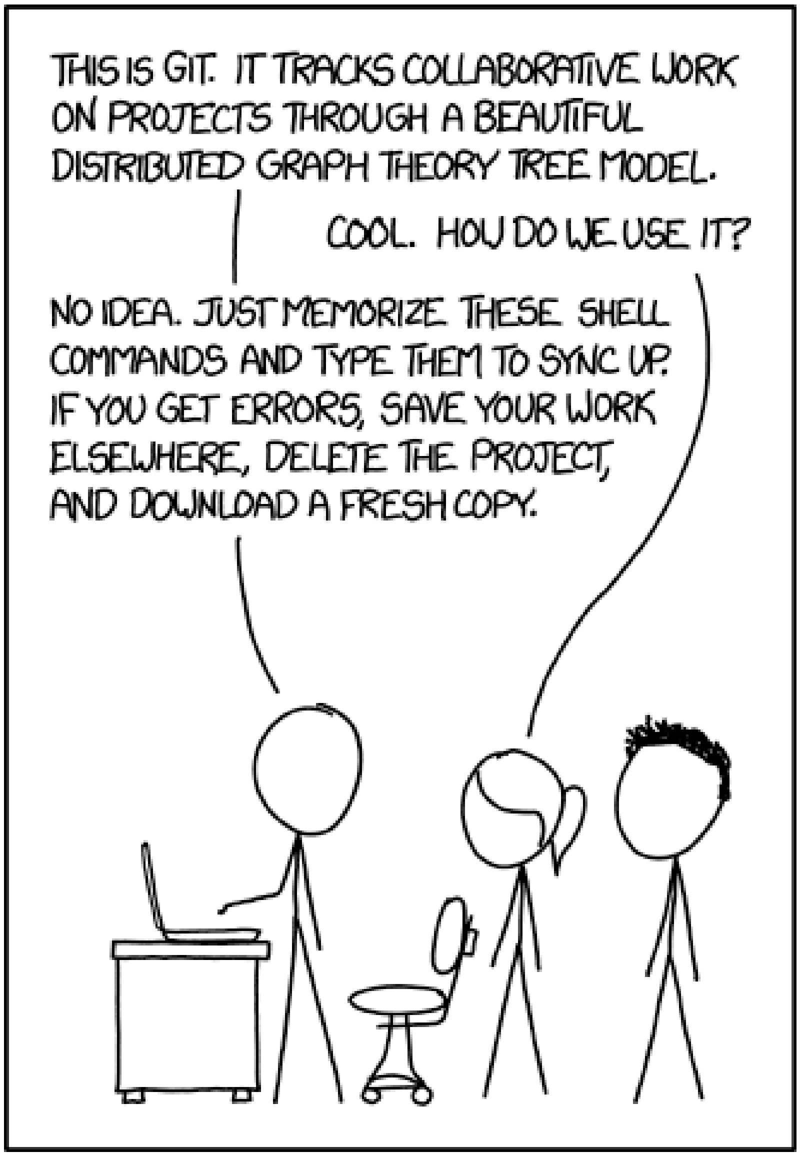


Figure 19.1: xkcd

Git commands are written `git verb`, where `verb` is what we actually want it to do. In this case, we're telling Git to configure our name and email address,

The two commands above only need to be run once: the flag `--global` tells Git to use the settings for every project on this machine.

19.2.1 Creating a repository

Once Git is configured, we can start using it to share code on GitHub.

Follow these instructions to create a new GitHub **repository**. Be sure to add these options:

1. Call the repo "plsc31101-final-project"
2. Create a README.md file
3. Create a .gitignore file
4. Don't add a license for now. Later, you can add a license for your project (see here for information on which license to choose.)

   Git without GitHub

   Git is often used in conjunction with GitHub. But you can also use git to track changes locally on your computer.

   If you wanted to start using Git from scratch on a new project, you can create a directory and tell Git to make it a repository – a place where Git can store old versions of our files – using the command `git init`

   ```
   $ git init
   ```

After you create your directory, **clone** a local copy onto your computer by following these instructions. Be sure to clone in a location that you will remember!

```
$ cd ~
$ git clone https://github.com/vlad/plsc31101-final-project.git
```

Now, navigate into your new git repository

```
$ cd ~/plsc31101-final-project
```

If we use `ls -a` to show the directory's contents, we can see a hidden directory called `.git`:

```
$ ls -a

# .                     .gitignore
# ..                    README.md
# .git
```

Git stores information about the project in this special sub-directory. If we ever delete it, we will lose the project's history.

We can check that everything is set up correctly by asking Git to tell us the status of our project:

```
$ git status
```

19.2.2 `git add`: tracks files

Let's add a file into our directory.

```
$ touch file.txt
```

Now, when we type in `git status`, we see something like this:

```
$ git status

# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
```

The "untracked files" message means that there's a file in the directory that Git isn't keeping track of. We can tell Git that it should do so using `git add`:

To add new files, you can either type `git add [file name]` like so:

```
$ git add file.txt
```

OR, if you want to add ALL the new files in a repository, you can use the `.` shortcut:

```
$ git add .
```

Now if we use `git status` we should no longer see any untracked files.

19.2.3 `git commit`: saves files

Git now knows that it's supposed to keep track of all the files in your repo,but it hasn't yet recorded any changes you've made to those files. To get it to do that, we need to run one more command:

```
$ git commit -am "First Commit"

# [master (root-commit) f22b25e] First Commit
 # 1 file changed, 1 insertion(+)
 # create mode 100644 ...
```

When we run `git commit`, Git takes everything we have told it to save by using `git add` and stores a copy permanently inside the special `.git` directory. This permanent copy is called a **revision** and its short identifier is `f22b25e`. (Your revision may have another identifier.)

We use the `-a` flag (for 'all') to tell Git that we want to commit all the changes we've made to every file. If we just run the `git commit` without the `-a` option, Git will expect us to specify which file's changes we want saved.

We use the `-m` flag (for "message") to record a comment that will help us remember later on what we did and why. If we just run `git commit` without the `-m` option, Git will launch `nano` (or whatever other editor we configured at the start) so that we can write a longer message.

If we run `git status` now:

```
$ git status

# On branch master
# nothing to commit, working directory clean
```

it tells us everything is up to date.

If we want to know what we've done recently, we can ask Git to show us the project's history using `git log`:

```
$ git log

# commit f22b25e3233b4645dabd0d81e651fe074bd8e73b
# Author: Vlad Dracula <vlad@tran.sylvan.ia>
# Date:   Thu Aug 22 09:51:46 2013 -0400

    # First commit
```

`git log` lists all revisions made to a repository in reverse chronological order. The listing for each revision includes the revision's full identifier (which starts with the same characters as the short identifier printed by the `git commit` command earlier), the revision's author, when it was created, and the log message Git was given when the revision was created.

19.2.4 `git push`: moves changes from one branch to another.

Systems like git allow us to move work between any two repositories. In practice, it's easiest to use one copy as a central hub, and to keep it on the web rather than on someone's laptop.

This is where GitHub comes in: it holds the **master** copy of a repository, and allows us to move changes between multiple **local** copies.

To copy our changes from our laptop to our GitHub repo, we can use `git push`:

```
$ git push origin master

# Counting objects: 9, done.
# Delta compression using up to 4 threads.
# Compressing objects: 100% (6/6), done.
# Writing objects: 100% (9/9), 821 bytes, done.
# Total 9 (delta 2), reused 0 (delta 0)
# To https://github.com/vlad/plsc31101-final-project
 # * [new branch]      master -> master
# Branch master set up to track remote branch master from origin.
```

This tells git to push our changes to the repository's "origin" – i.e., the copy on GitHub.

Now open up a web browser and navigate to your GitHub repository. What do you see?

19.2.5 Challenge 1

Navigate to https://github.com/plsc-31101/replication-template and clone the repository to your computer.

Copy all the files and directories in this folder into your new github repo (plsc31101-final-project).

Then add, commit, and push. Use this template for your final project!

Cheat sheet

```
$ git add .
$ git commit -am "commit message"
$ git push origin/master
```

19.2.6 Ignoring Things

Oftentimes we'll have files that we do not want git to track for us. These include sensitive data files, as well as hidden files with extensions like .Rhistory, .ipynb_checkpoints, and .DS_Store (Dropbox).

Let's create a few dummy files:

```
$ touch a.dat b.dat data/c.csv data/d.csv
```

and see what Git says:

```
$ git status

# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   a.dat
#   b.dat
#   data/
# nothing added to commit but untracked files present (use "git add" to track)
```

Putting these files under version control would be a waste of disk space. What's worse, having them all listed could distract us from changes that actually matter, so let's tell Git to ignore them.

We do this by creating a file in the root directory of our project called `.gitignore`.

```
$ nano .gitignore
$ cat .gitignore

*.dat
data/
```

These patterns tell Git to ignore any file whose name ends in `.dat` and everything in the `data` directory. (If any of these files were already being tracked, Git would continue to track them.)

Once we have created this file, the output of `git status` is much cleaner:

```
$ git status

# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#   .gitignore
# nothing added to commit but untracked files present (use "git add" to track)
```

The only thing Git notices now is the recently-changed `.gitignore` file. You might think we wouldn't want to track it, but everyone we're sharing our repository with will probably want to ignore the same things that we're ignoring. Let's add and commit `.gitignore`:

```
$ git commit -m "Add the ignore file"
$ git status

# On branch master
```

```
# nothing to commit, working directory clean
```

As a bonus, using `.gitignore` helps us avoid accidentally adding files to the repository that we don't want.

```
$ git add a.dat

# The following paths are ignored by one of your .gitignore files:
# a.dat
# Use -f if you really want to add them.
# fatal: no files added
```

If we really want to override our ignore settings, we can use `git add -f` to force Git to add something. We can also always see the status of ignored files if we want:

```
$ git status --ignored

# On branch master
# Ignored files:
#  (use "git add -f <file>..." to include in what will be committed)
#
#        a.dat
#        b.dat
#        c.dat
#        results/

# nothing to commit, working directory clean
```

19.2.7 Challenge 2

Continue editing the .gitignore file to add extentions you don't want to track, like .DS_Store, .Rhistory, etc.

19.2.8 Pulling / Syncing

Oftentimes we need to sync our local repo with the *master branch* (the default branch) on GitHub. For instance, let's say you have two computers, one at home and one at work. We made a change using our work computer, and pushed it to the master branch on GitHub. But then we go home and find that our local copy is out of date.

A more common method of syncing branches is to use `git fetch` followed by `git merge`; or `git pull`.

1. `git fetch` followed by `git merge` *combines* your local changes with changes made by others.
2. `git pull` is a convenient shortcut for completing both `git fetch` and `git merge` in the same command

```
$ git pull origin/master
```

This is helpful if you want to merge your changes and the master branch.

> **Commit before your pull**
>
> Because `pull` performs a merge on the retrieved changes, you should ensure that your local work is committed before running the pull command. If you run into a merge **conflict** you cannot resolve, or if you decide to quit the merge, you can use `git merge --abort` to take the branch back to where it was in before you pulled. See here for more info on fetching and merging.

Sometimes `git pull` will cause **merge conflicts**, meaning that your local repository and master branch diverged on some lines of code, and git doesn't know which version you want to keep.

If you *know* that you want to keep the master branch version, you can overwrite your local repository like this:

```
$ git fetch
$ git reset --hard origin/master
```

Let's break this down:

1. `git fetch` retrieves a record of all changes made in the master branch.
2. `git reset --hard origin/master` will reset your local repo to match the master branch.

With these commands, every tracked file will be overwritten to match to its version in master. Be careful with this: all local changes will be lost.

19.3 Collaborating

Version control really comes into its own when we begin to collaborate with other people.

All of the course notes are contained in their own Github Repo: https://github.com/plsc-31101/course

I've created a directory in this repo called `98-Final-Projects`. We're going to collaborate on this directory using git to collect information about your final projects.

There are two main ways to collaborate on github:

1. Adding individual collaborators to a project
2. The **fork & pull** model.

The first method adds users to your project, giving them full permissions to make changes. For this course, I added Pete Cuppernull as a collaborator, so that he could push commits easily to the repository without my expressed approval.

Collaborating in this fashion is very similar to the workflow described above.

19.3.1 Fork & Pull Model

GitHub also allows you to to accept individual contributions from users without granting them full access. This if referred to as the *Fork & Pull model.*

Fork & Pull involves the following steps:

1. Fork an existing repo

The first step in in this workflow is to **fork** an existing repository. A fork is a copy of a repository that you manage yourself. Forks let you make changes to a project without affecting the original repository.

To fork a repo:

1. On GitHub, navigate to plsc-31101/course,
2. In the top-right corner of the page, click **Fork**.

Now you have a fork of the original repo in *your-user-name*/course

2. Commit a change

We've already seen how you can commit a change directly in GitHub's web interface. But when working with code, you often want to develop your scripts on your computer, so you can test it using R, Python, etc.

To do this, you first need to **clone your fork** onto your computer.

1. On GitHub, navigate to your fork of the `course` repository.
2. In the right sidebar of your fork's repository page, copy the clone URL for your fork.
3. Use `git clone` to clone the repo.

```
$ cd ~
$ git clone https://github.com/YOUR-USERNAME/course
```

We're now ready to make a change to the repo. Create a file in `98-Final-Projects` directory named after yourself. **Protip**: the `touch` command quickly creates an empty file.

```
$ cd course
$ touch 98-Final-Projects/rochelle-terman.md
```

Open up that file in a text editor (I like Sublime Text) and add:

1. The title of your project (in a header 2)
2. A 1-2 sentence description of the project
3. A link to your github repo (that you just made)

> **Markdown** Reminder
>
> Files with the extension .md are called **markdown** files. Markdown is a markup language used to convert plain text to HTML and many other formats. It's basically a way to add markup to a text (making things bold, lists, links, etc) using very simple syntax. It is often used in README files in software packages. You may have also noticed that all of the lessons for this course are written in markdown, as is many of the text files on GitHub. You can learn more about how to write GitHub-flavored markdown here.

Then add, commit, and push the change.

```
$ git status
$ git add 16_final-projects/rochelle-terman.md
$ git commit -am "my final project info"
$ git push
```

3. Submit a pull request

Navigate to your GitHub repo (online) and check out your change!

Remember when you forked the repository originally? That means that your repository is different from mine, and from everybody elses. What if you want to share your change with others?

To do this, navigate to your GitHub repository and click the green icon to submit a **pull request**.

After you submit, I have the option to accept.

4. Keep your fork synced

It's good practice to keep your fork synced with the upstream (i.e. the original) repository. That way, if I make a change to PS239T, you can easily pull that change into your fork.

You can configure Git to pull changes from the original, or upstream, repository into the local clone of your fork.

```
$ git remote -v
$ git remote add upstream https://github.com/plsc-31101/course.git
$ git remove -v
```

With `git remote -v`, you'll see the current configured remote repository for your fork.

Now you can sync your fork with the upsteam repo using `git fetch`:

```
$ git fetch upstream
$ git merge upstream/master
```

To learn more:

1. [GitHub documentation](https://help.github.com
2. This great stackoverflow answer

Part III

Assignments and Projects

Chapter 20

Assignments

20.1 Assignment 1 Solutions

- **Assigned**: Oct 3, 2019.
- **Due**: Oct 10, 2019 at 12:29pm.

For this assignment, you will confirm that everything is installed and setup correctly, and you understand how to interact with R Studio and R Markdown.

Your answers (to this assignment only) will be posted on our course website.

1. Using R Markdown

In the space below, insert a picture of yourself, and complete the following information:

1. **Name**: Daenerys Targaryen
2. **Department and degree program**: Queen of the Andals and the First Men, Protector of the Seven Kingdoms, the Mother of Dragons, the Khaleesi of the Great Grass Sea, the Unburnt, the Breaker of Chains.
3. **Year in the program**: First.
4. **One-sentence description of academic interests**: I am interested in slavery, intercontinental conflict, and pyrology.
5. **Some non-academic interests**: Dragons, Jon Snow, eating raw hearts.
6. **R version installed on your computer** (Open a command line window ('terminal' or, on windows, 'git bash'), and enter the following command `R --version`): 3.6.1
7. **R Studio version installed on your computer** (Open RStudio and, in the navigation menu, click on RStudio –> About RStudio): 1.1.456
8. **Primary computer operating system** (Mac OS, Windows, Linux, etc): Mac OS 10.13.6.
9. **Programming experience** (How would you describe your previous programming experience?): None.

2. Checking packages

Create an R chunk below, where you load the `tidyverse` library.

```
library(tidyverse)
```

3. Knit and submit.

Knit the R Markdown file to PDF. Submit **BOTH** the .Rmd file and the PDF file to Canvas.

If you get an error trying to knit, read the error and make sure that your R code is correct. If that doesn't work, confirm you've correctly installed the requisite packages (`knit`, `rmarkdown`). If you still can't get it to work, paste the error on Canvas.

20.2 Assignment 2 Solutions

- **Assigned**: Oct 10, 2019.
- **Due**: Oct 17, 2019 at 12:29pm.

For this assignment, you'll use what you know about R syntax and data structures to perform some common data operations.

1. Basics

1.1 Fix the following syntax errors. Enter your corrected code in the second chunk.

```
# 1
states <- ("California", "Illinois", "Ohio")

# 2
countries <- c("Iran", "Indonesia," "India", "Italy")

# 3
df <- data.frame(age = c(21, 66, 35)
                 party = c('rep', 'dem', 'rep'))

# 4
my-vector <- c("apples", "oranges", "kiwis")

# 5
artists <- list(names = c("Picasso", "Kahlo",
                genre = c("cubist", "surrealist"))
```

```
# PUT YOUR CORRECTED CODE HERE

# 1
states <- c("California", "Illinois", "Ohio")
```

```r
# 2
countries <- c("Iran", "Indonesia", "India", "Italy")

# 3
df <- data.frame(age = c(21, 66, 35),
                 party = c('rep', 'dem', 'rep'))

# 4
my_vector <- c("apples", "oranges", "kiwis")

# 5
artists <- list(names = c("Picasso", "Kahlo"),
                genre = c("cubist", "surrealist"))
```

1.2 How many arguments does the `order()` function pass? What are they?

2. Vectors and Lists

2.1 Create three vectors:

- a character vector, `titles`, that contain the names of 3 of your favorite movies
- a numeric vector, `year`, that contains the years in which those movies were produced
- a boolean vector `bechdel` that TRUE/FALSE according to whether those movies pass the bechdel test

```r
titles <- c("Dog Day Afternoon", "The Graduate", "Breakfast Club")
year <- c(1975, 1967, 1985)
bechdel <- c(TRUE, FALSE, TRUE)
```

2.2 Put those three vectors in a list, called `movies`.

```r
movies <- list(titles, year, bechdel)
```

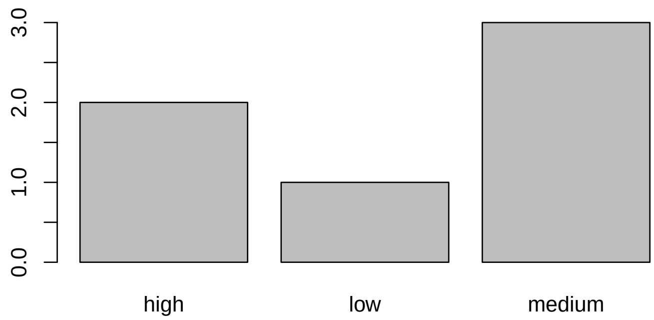
2.3 Print the structure of the list `movies`.

```r
str(movies)
#> List of 3
#>  $ : chr [1:3] "Dog Day Afternoon" "The Graduate" "Breakfast Club"
#>  $ : num [1:3] 1975 1967 1985
#>  $ : logi [1:3] TRUE FALSE TRUE
```

3. Factors

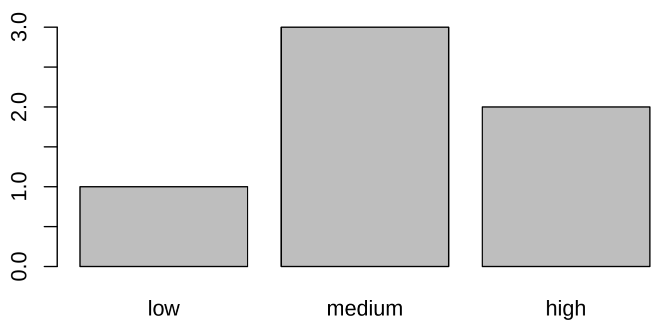
3.1 Here's some code that prints a simple barplot:

```
f <- factor(c("low","medium","high","medium","high","medium"))
table(f)
#> f
#>   high    low medium 
#>      2      1      3
barplot(table(f))
```



How would you relevel f to be in the correct order?

```
f <- factor(f, levels = c("low", "medium", "high"))

# Test your code
barplot(table(f))
```



4. Dataframes

4.1 Coerce the movies object you made above from a list to a dataframe. Call it movies_df.

```
movies_df <- as.data.frame(movies)
```

4.2 Add appropriate column names to movies_df.

```r
names(movies) <- c("film", "year", "bechtel")
```

20.3 Assignment 3 Solutions

- **Assigned**: Oct 17, 2019.
- **Due**: Oct 24, 2019 at 12:29pm.

For this assignment, you'll be working on some real life data! I've prepared for your a basic country-year dataset, with the following variables:

- Country name
- Country numerical code
- Year
- UN Ideal point
- Polity2 score of regime type (from Polity VI)
- Physical Integrity Rights score (from CIRI dataset)
- Speech Rights score (from CIRI)
- GDP per capita (from World Bank)
- Population (from World Bank)
- Political Terror Scale using Amnesty International reports (from Political Terror Scale project)
- Composite Index of Military Capabilities (Correlates of War)
- Region

1. R Projects and Importing

1.1 Using `getwd()`, print your working directory below.

```r
getwd()
#> [1] "/Users/rterman/Desktop/course"
```

1.2 Read `country-year.csv` into R, using a relative path. Store it in an object called `dat`.

```r
dat <- read.csv("data/country-year.csv")
```

2. Dimensions and Names

2.1 How many rows and columns are in the dataset?

```r
dim(dat)
#> [1] 6416   13
```

2.2 Print the column names.

```
names(dat)
#>  [1] "X"          "year"       "ccode"      "country"    "idealpoint"
#>  [6] "polity2"    "physint"    "speech"     "gdp.pc.wdi" "pop.wdi"
#> [11] "amnesty"    "cinc"       "region"
```

2.3 Remove the `X` column from the dataset.

```
dat$X <- NULL
```

2.4 One of the variables is called "gdp.pc.wdi". This stands for "Gross Domestic Product Per Capita, from the World Bank Development Indicators". Change this variable name in the dataset from " "gdp.pc.wdi" to "GDP"

```
names(dat)[8] <- "GDP"
```

3. Summarizing

3.1 How many years are covered in the dataset?

```
length(unique(dat$year))
#> [1] 36
```

3.2 How many unique countries are covered in the dataset?

```
length(unique(dat$country))
#> [1] 196
```

3.3 What is the range of `polity2`? How many NAs are in this column?

```
summary(dat$polity2)
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
#>     -10      -6       5       2       9      10    1214
```

4. Subsetting

4.1 Subset dat so that it returns the third column AS A VECTOR (Do not print the object; store it in a variable.)

```
sub <- dat[[3]]
#OR
sub <- dat[,3]
#OR
names(dat)[3]
#> [1] "country"
sub <- dat$country
```

4.2 Fix each of the following common data frame subsetting errors:

1. Extract observations collected for the year 1980

```
dat[dat$year = 1980,]

# Corrected
dat[dat$year == 1980,]
```

2. Extract all columns except 1 through to 4

```
dat[,-1:4]

# Corrected
dat[,-c(1:4)]
```

3. Extract the rows where the polity2 score is greater than 5

```
dat[dat$polity2 > 5]

# corrected
dat[dat$polity2 > 5, ]
```

4. Extract the first row, and the third and fourth columns (`country` and `idealpoint`).

```
dat[1, 3, 4]

# Corrected
dat[1, c(3, 4)]
```

5. Extract rows that contain information for the years 2002 and 2007

```
dat[dat$year == 2002 | 2007,]

# Corrected
dat[dat$year == 2002 | dat$year == 2007,]
```

4.3 What does `summary(dat$polity2[dat$region =="Africa"])` do? Explain below in your own words.

It calculates some summary statistics for `polity2` scores from observations in Africa.

4.4 Subset the data to include only observations from years 1990-2000 (inclusive). Put the subsetted data in a new variable called `dat.1990.2000`

```
dat.1990.2000 <- dat[dat$year >= 1990 & dat$year<=2000,]
```

4.5 Using `mean()` function, tell me the average GDP of observations from 1990 to 2000.

```
mean(dat.1990.2000$GDP, na.rm = T)
#> [1] 6611
```

4.6 You just calculated the average GDP for years 1990-2000. Now calculate the average GDP from 2001 onwards. Tell me how much larger it is (in percentage).

```
dat.2001.plus <- dat[dat$year > 2000,]
mean1 <- mean(dat.1990.2000$GDP, na.rm = T)
mean2 <- mean(dat.2001.plus$GDP, na.rm = T)
(mean2 - mean1) / mean1
#> [1] 0.825
```

4.7 Look up the helpfile for the function `is.na()`. Using this function, replace all the NA values in the `polity2` column of `dat` with 0.

```
?is.na
dat$polity2[is.na(dat$polity2)] <- 0
summary(dat$polity2)
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#>  -10.00   -5.00    0.00    1.46    8.00   10.00
```

20.4 Assignment 4 Solutions

- **Assigned**: Oct 24, 2019
- **Due**: Nov 5, 2019 at 12:29pm.

For this problem set, we'll be working with the country-year data introduced in the last assignment. As a reminder, the dataset contains the following variable:

- `year`: Year.
- `ccode`: Country numerical code.
- `country`: Country name.
- `idealpoint` UN Ideal point.
- `polity2`: Polity2 score of regime type (from Polity VI).
- `physint`: Physical Integrity Rights score (from CIRI dataset).
- `speech`: Speech Rights score (from CIRI).
- `gdp.pc.wdi`: GDP per capita (from World Bank).
- `pop.wdi`: Population (from World Bank).
- `amnesty`: Political Terror Scale using Amnesty International reports (from Political Terror Scale project).
- `cinc`: Composite Index of Military Capabilities (Correlates of War).
- `region`: Geographic region.

We'll be merging this country_year data with new data about U.S. news coverage of women around the world (excluding the United States). In this new dataset, the unit of observation is *article*. That is, each row represents an individual article, with columns for:

- `publication`: NYT or Washington Post.
- `year`: Year article was published.
- `title`: Title of the article.
- `country`: Country the article is mainly about.
- `region`: Region where `country` is located.
- `ccode`: Numerical code for `country`.

1. Loading, subsetting, summarizing

1.1 Load the csv found in `data/articles.csv` into R. Be sure to set `stringsAsFactors` to `FALSE`. Store the data-frame to an object called `articles` and tell me the variable names.

```
library(tidyverse)
articles <- read.csv("data/articles.csv", stringsAsFactors = F)
names(articles)
#> [1] "publication" "year"        "title"       "country"     "region"
#> [6] "ccode"
```

1.2 How many countries are covered in the dataset?

```
length(unique(articles$country))
#> [1] 147
```

1.3 The variable `ccode` reports a numerical ID corresponding to a given country. Print the names of the country or countries without a `ccode` (i.e. those countries where the `ccode` is `NA`.)

```
unique(articles$country[is.na(articles$ccode)])
#> [1] "Palestine"
```

1.4 Remove all articles where the `ccode` variable is `NA`. How many observations are left with?

```
articles_no_na <- articles[!is.na(articles$ccode), ]
nrow(articles_no_na)
#> [1] 4494
```

2. Counting Frequencies and Merging

2.1 Create a new data frame called `articles_country_year` that tells us the number of articles per `ccode` (i.e. country code), per `year`.

The final data frame `articles_country_year` should contain three columns: `ccode`, `year`, and `number_articles`.

Print the first 6 rows of the `articles_country_year`.

Hint: The `count` function – part of the `plyr` package – might be helpful.

```
articles_country_year <- articles_no_na %>%
  dplyr::count(ccode, year) %>%
  select(ccode, year, number_articles = n)

kable(head(articles_country_year))
```

ccode

year

number_articles

20

1980

4

20

1981

9

20

1982

4

20

1983

4

20

1984

6

20

1985

1

2.2. Load `data/country-year.csv` (this is the country-year data we worked with during the last assignment.)

```
country_year <- read.csv("data/country-year.csv", stringsAsFactors = F)
```

2.3 Subset `country_year` such that it has the same year range as `articles_country_year`.

```
range(articles_country_year$year)
#> [1] 1980 2014
range(country_year$year)
#> [1] 1979 2014

country_year <-country_year %>%
  filter(year > 1979)
```

2.4 Merge (i.e. join) `articles_country_year` and `country_year` into a new dataframe called `merged`.

When you're done, `merged` should have all the rows and columns of the country__year dataset, along with a new column called `number_articles`.

Print the first 6 rows of this new dataframe `merged`.

```
merged <- country_year %>%
  left_join(articles_country_year)
#> Joining, by = c("year", "ccode")

kable(head(merged))
```

X

year

ccode

country

idealpoint

polity2

physint

speech

gdp.pc.wdi

pop.wdi

amnesty

cinc

region

number_articles

156

1980

700

Afghanistan

-1.560

NA

NA

NA

276

13180431

5

0.001

MENA

NA

157

1980

540

Angola

-1.176

-7

NA

NA

NA

7637141

3

0.001

Africa

NA

158

1980

339

Albania

-1.564

-9

NA

NA

NA

2671997

3

0.001

EECA

NA

159

1980

696

United Arab Emirates

-0.315

-8

NA

NA

42962

1014825

NA

0.001

MENA

NA

160

1980

160

Argentina

0.128

-9

NA

NA

2737

28120135

5

0.007

LA

NA

161

1980

900

Australia

1.423

10

NA

NA

10188

14692000

NA

0.007

West

3

2.5 In `merged`, replace all instances of `NA` in the `number_articles` column to 0.

```
# solution 1 - base R
merged$number_articles[is.na(merged$number_articles)] <- 0

# solution 2 - tidyr
merged$number_articles <- replace_na(merged$number_articles, 0)

# solution 3 - dplyr
merged <- merged %>%
  mutate(number_articles = ifelse(is.na(number_articles), 0, number_articles))

# test
```

```
summary(merged$number_articles)
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#>     0.0     0.0     0.0     0.7     0.0    99.0
```

2.6 Which country-year observation has the most number of articles? Write code that prints the year, country name, and number of articles for this observation.

```
# solution #1 -- base R
merged[which.max(merged$number_articles),c("year", "country", "number_articles")]
#>      year country number_articles
#> 5950 2013   India              99

# solution #2 -- tidyverse
merged %>%
  top_n(1, number_articles) %>%
  select(year, country, number_articles)
#>   year country number_articles
#> 1 2013   India              99
```

3. Group-wise Operations

3.1 Using the `merged` data and our split-apply-combine strategies, print the total number of articles about women per region.

```
n_region <- merged %>%
  group_by(region) %>%
  summarise(count = sum(number_articles, na.rm = T))

n_region
#> # A tibble: 7 x 2
#>   region count
#>   <chr>  <dbl>
#> 1 Africa   464
#> 2 Asia    1288
#> 3 EECA     251
#> 4 LA       328
#> 5 MENA     940
#> 6 West    1159
#> # ... with 1 more row
```

4. Long v. wide formats

4.1 Create a piped operation on `merged` that does the following:

1. Subsets the dataframe to select `year`, `country`, and `number_articles` columns. 2. Filters the dataframe to select only observations in the MENA region. 3. Spreads the dataframe so that each `country` is a column, and the cells represent 'number__articles.

Print the first 6 rows of this transformed data frame.

```
wide <- merged %>%
  filter(region == "MENA") %>%
  select(year, country, number_articles) %>%
  spread(country, number_articles, fill = 0)

kable(head(wide))
```

year

Afghanistan

Algeria

Bahrain

Egypt

Iran

Iraq

Israel

Jordan

Kuwait

Lebanon

Libya

Morocco

Oman

Palestine

Qatar

Saudi Arabia

South Sudan

Sudan

Syria

Tunisia

Turkey

United Arab Emirates

Yemen Arab Republic

Yemen People's Republic

1980

0

0

0

0

4

0

9

0

0

0

0

0

0

0

0

1

0

0

0

0

0

0

0

0

1981

1

0

0

2

1

0

2

0

0

0

0

1

0

0

0

1

0

0

0

0

1

0

0

0

1982

0

0

0

1

2

1

6

0

1

0

0

0

0

0

0

1

0

0

0

0

1

0

0

0

1983

0

0

0

0

0

0

2

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

0

1984

0

0

0

3

0

0

3

0

1

2

0

0

0

0

0

0

0

0

1

0

0

0

0

0

1985

0

0

0

6

0

0

1

0

1

3

0

0

0

0

0

0

0

0

0

0

0

0

0

0

4.2 Transform the dataset you created above back into `long` format, with three variables: `year`, `country`, and `number_articles`

Print the first 6 rows of this transformed data frame.

```
long <- wide %>%
  gather(country, number_articles, -year)

kable(head(long))
```

year

country

number_articles

1980

Afghanistan

0

1981

Afghanistan

1

1982

Afghanistan

0

1983

Afghanistan

0

1984

Afghanistan

0

1985

Afghanistan

0

20.4.0.1 Extra Credit

This question is not required. But it you want an extra challenge....

Transform the `country_year` data into an undirected dyadic dataset. Here, the unit of observation should be the dyad-year, with five columns:

1. `ccode_1`: Country 1 `ccode`
2. `country_1`: Country 1 name
3. `ccode_2`: Country 2 `ccode`
4. `country_2`: Country 2 name
5. `year`: Year of observation
6. `gdp_diff`: Absolute difference of gdp between dyad.

This should be *undirected* dyadic dataset, meaning `USA-Canada-1980` is the same as `Canada-USA-1980`, and we shouldn't have duplicate rows for the same dyad.

Try to do it all in 1 piped sequence. Then tell me the dyad-year with the greatest wealth disparity.

```
dyad <- country_year %>%
  expand(ccode_1=ccode, ccode_2=ccode) %>% # make two columns of states
  filter(ccode_1 > ccode_2) %>% # from directed to undirected dyads
  left_join(., country_year, by=c("ccode_1"="ccode")) %>% # get state1 info
  left_join(., country_year, by=c("year", "ccode_2"="ccode")) %>% # get state2 info
  mutate(gdp_diff = abs(gdp.pc.wdi.x - gdp.pc.wdi.y)) %>% # take absolute difference i
  select(ccode_1, country_1 = country.x, ccode_2, country_2 = country.y, year, gdp_dif
  arrange(desc(gdp_diff))

kable(head(dyad))
```

ccode_1

country_1

ccode_2

country_2

year

gdp_diff

516

Burundi

221

Monaco

2008

193705

450

Liberia

221

Monaco

2008

193661

531

Eritrea

221

Monaco

2008

193636

553

Malawi

221

Monaco

2008

193590

490

Democratic Republic of the Congo

221

Monaco

2008

193566

530

Ethiopia

221

Monaco

2008

193565

20.5 Assignment 5 Solutions

- **Assigned**: Nov 5, 2019

- **Due**: Nov 12, 2019 at 12:29pm.

For this problem set, we'll be working with the country-year data introduced in the last assignment. As a reminder, the dataset contains the following variables:

- `year`: Year.
- `ccode`: Country numerical code.
- `country`: Country name.
- `idealpoint` UN Ideal point.
- `polity2`: Polity2 score of regime type (from Polity IV).
- `physint`: Physical Integrity Rights score (from CIRI dataset).
- `speech`: Speech Rights score (from CIRI).
- `gdp.pc.wdi`: GDP per capita (from World Bank).
- `pop.wdi`: Population (from World Bank).
- `amnesty`: Political Terror Scale using Amnesty International reports (from Political Terror Scale project).
- `cinc`: Composite Index of Military Capabilities (Correlates of War).
- `region`: Geographic region.

1. Getting Started

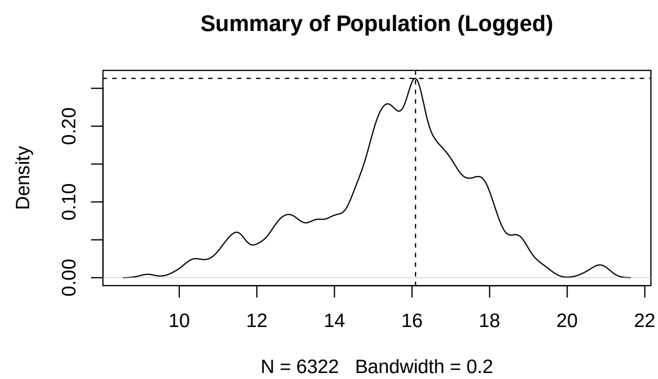
1.1 Read `data/country-year.csv` into R, using a relative path. Store it in an object called `dat`.

```
library(tidyverse)
library(stargazer)
dat <- read.csv("Data/country-year.csv", stringsAsFactors = F)
```

2. Plotting

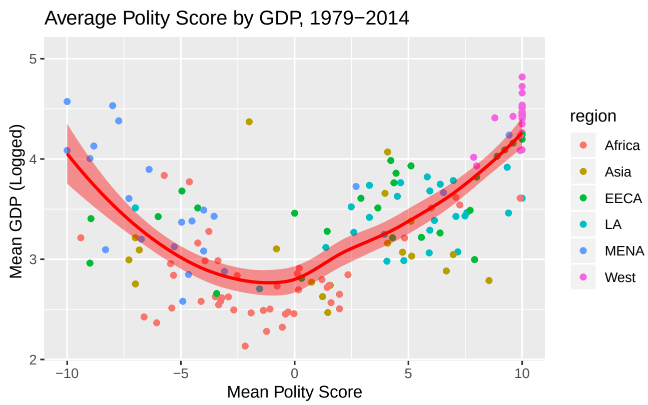
2.1 Write code that reproduces "plots/Plot_1.jpeg". (No need to write the file.)

```
# Density of population
d <- density(log(dat$pop.wdi), na.rm = T, bw = .2)
plot(d, main = "Summary of Population (Logged)")
abline(h = max(d$y), v = 16.09, lty = 2)
```



2.2 Write code that reproduces "plots/Plot_2.jpeg". (No need to write the file.)

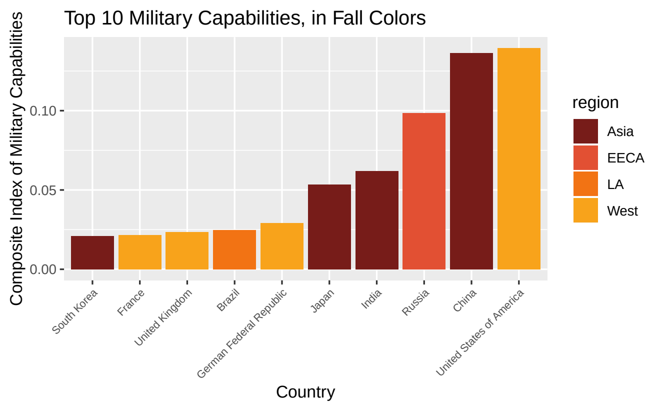
```
# get summary data
country_means <- dat %>%
  filter(!is.na(region)) %>%
  group_by(country) %>%
  summarise(gdp = mean(gdp.pc.wdi, na.rm = T),
            polity = mean(polity2, na.rm = T),
            cinc = mean(cinc, na.rm = T),
            region = region[1])

# plot
ggplot(country_means, aes(x = polity, y = log10(gdp))) +
  geom_point(aes(color = region)) +
  #scale_y_log10() +
  geom_smooth(color="red", fill="red") +
  ylab("Mean GDP (Logged) ") +
  xlab("Mean Polity Score") +
  ggtitle("Average Polity Score by GDP, 1979-2014")
#> `geom_smooth()` using method = 'loess' and formula 'y ~ x'
#> Warning: Removed 33 rows containing non-finite values (stat_smooth).
#> Warning: Removed 33 rows containing missing values (geom_point).
```



2.3 Write code that reproduces "plots/Plot_3.jpeg". (No need to write the file.)

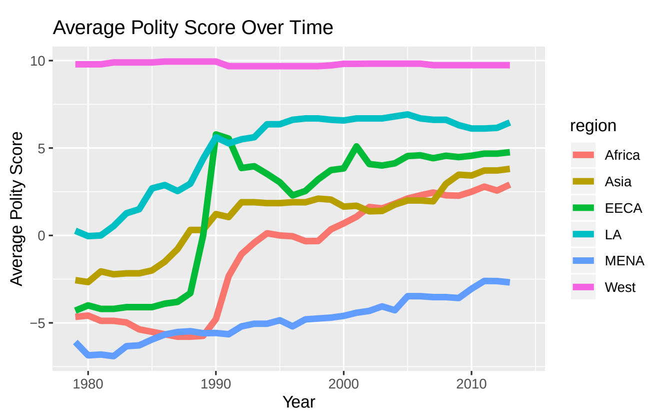
Hint: The fall-inspired colors are #771C19, #E25033, #F27314, #F8A31B

```
# military capabilities
top_cinc <- country_means %>%
  top_n(10, cinc)

# Fall theme
rhg_cols = c("#771C19","#E25033","#F27314", "#F8A31B")

# plot
ggplot(top_cinc, aes(reorder(country, cinc), cinc, fill = region)) +
  geom_col() +
  theme(axis.text.x=element_text(size = 7, angle=45, hjust=1)) +
  ylab("Composite Index of Military Capabilities") +
  xlab("Country") +
  ggtitle("Top 10 Military Capabilities, in Fall Colors") +
  scale_fill_manual(values = rhg_cols)
```



2.4 Write code that reproduces "plots/Plot_4.jpeg". (No need to write the file.)

```
# prepare data
year_means <- dat %>%
  filter(!is.na(region)) %>%
  group_by(year, region) %>%
  summarise(gdp = mean(gdp.pc.wdi, na.rm = T),
            polity = mean(polity2, na.rm = T),
            physint = mean(physint, na.rm = T))

# plot
ggplot(year_means, aes(x = year, y = polity, color = region)) +
  geom_line(size=2) +
  ylab("Average Polity Score") +
  xlab("Year") +
  ggtitle("Average Polity Score Over Time")
#> Warning: Removed 6 rows containing missing values (geom_path).
```



3. Models

3.1 Write code that reproduces the model summary table "reg_table.txt" (and writes the file).

```
mod.1 <- lm(physint ~ polity2, data = dat)
mod.2 <- lm(physint ~ polity2 + log(gdp.pc.wdi), data = dat)
mod.3 <- lm(physint ~ polity2 + log(gdp.pc.wdi) + region, data = dat)

stargazer(mod.1, mod.2, mod.3, title = "Regression Results", type = "text",
          covariate.labels  = c("Polity2", "GDP per capita, logged", "Asia", "Eastern E
          dep.var.labels = "DV: Physical Integrity",
          omit = "Constant",
          keep.stat="n", style = "ajps",
          out = "reg_table.txt")
#>
#> Regression Results
#> --------------------------------------------------
#>                          DV: Physical Integrity
#>                        Model 1  Model 2   Model 3
#> --------------------------------------------------
#> Polity2                0.124*** 0.064*** 0.033***
#>                        (0.004)  (0.005)   (0.005)
#> GDP per capita, logged          0.503*** 0.493***
#>                                 (0.021)   (0.028)
#> Asia                                     -1.170***
#>                                           (0.101)
#> Eastern Europe                            -0.033
#>                                           (0.109)
#> Latin America                            -0.769***
#>                                           (0.097)
#> MENA                                     -1.460***
#>                                           (0.116)
#> West                                     0.604***
#>                                           (0.138)
#> N                        4336     4141     4141
#> --------------------------------------------------
#> ***p < .01; **p < .05; *p < .1
```

Chapter 21

Final Projects

Part IV

Resources

Chapter 22

Cheat Sheets and Guides

1. RStudio IDE Cheat Sheet
2. R Markdown Cheat Sheet
3. R Markdown Reference Guide Base R Cheat Sheet
4. Data Transformation with `dplyr` Cheat Sheet
5. Data Visualization Cheat Sheet
6. Regular Expressions Cheat Sheet