Reproducible research tools to increase efficiency and reduce errors

Dominic LaRoche

May 4, 2015

Outline

- Introduction and Motivation
- The Tools of the Trade
- Examples
- Benefits to You and HTG

Section 1

Introduction

What is Reproducible Research?

What does it mean to have "reproducible" research?

What is Reproducible Research?

What does it mean to have "reproducible" research?

We break research down into 2 phases:

What is Reproducible Research?

What does it mean to have "reproducible" research?

We break research down into 2 phases:

1.) Data generating experiment
- Can the same data be generated again?
- This is an ongoing topic in the annals of *Nature* and *Science*

What is Reproducible Research?

What does it mean to have "reproducible" research?

We break research down into 2 phases:

1.) Data generating experiment
- Can the same data be generated again?
- This is an ongoing topic in the annals of *Nature* and *Science*

2.) Can the analysis and results be reproduced given the same data?
- The focus of my talk today

What is Reproducible Research?

What does it mean to have "reproducible" research?

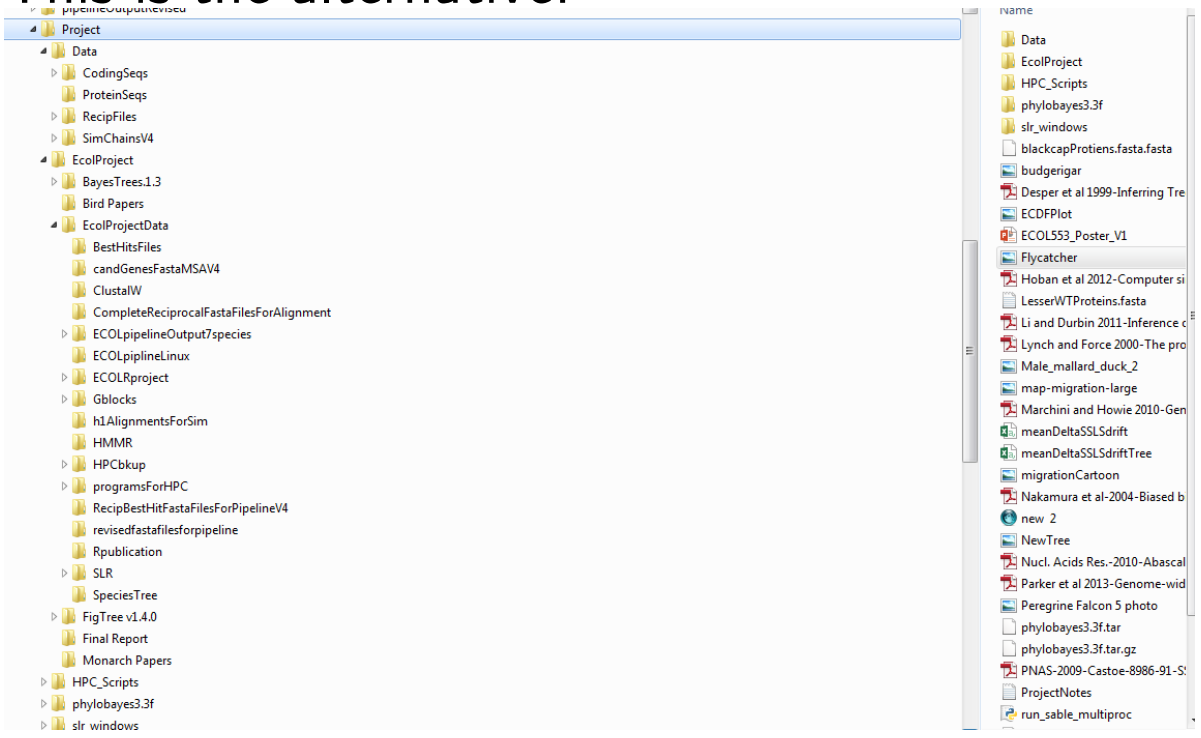
We break research down into 2 phases:

1.) Data generating experiment
- Can the same data be generated again?
- This is an ongoing topic in the annals of *Nature* and *Science*

2.) Can the analysis and results be reproduced given the same data?
- The focus of my talk today
- Three questions:
  1.) Can you reproduce the results right now?
  2.) Can you reproduce the results a year from now?
  3.) Can someone else reproduce the results in 5 years?

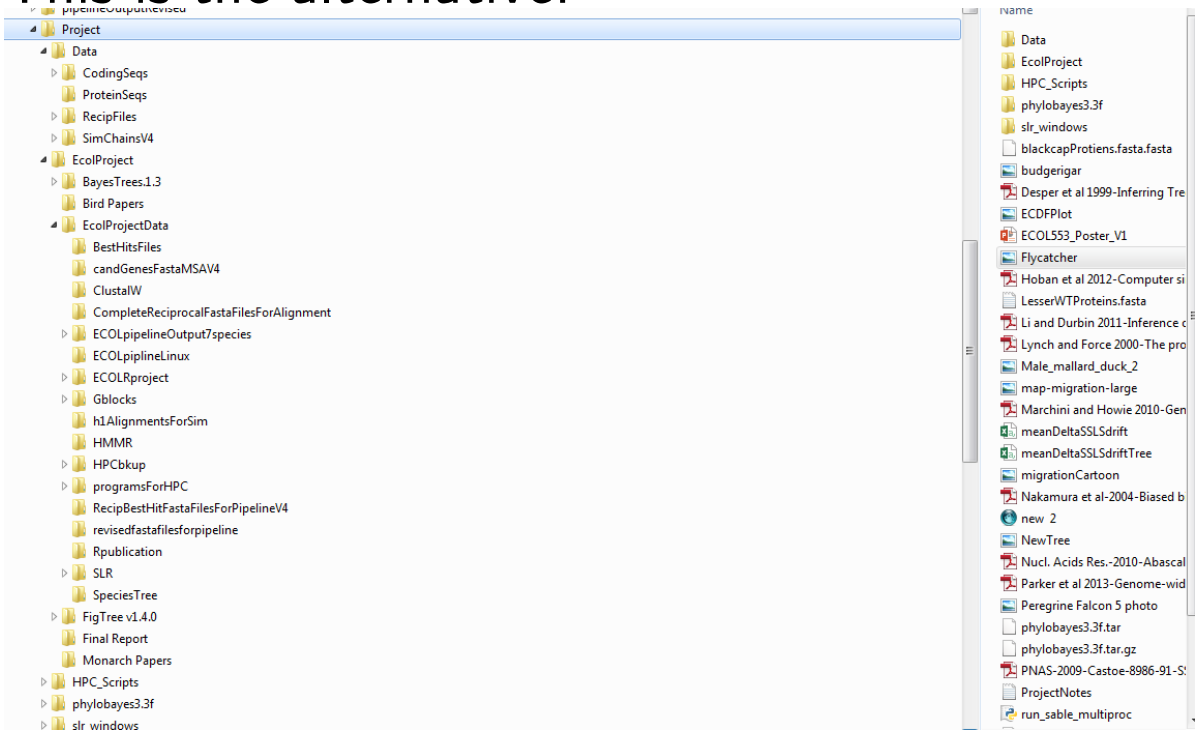
Why Reproducible Research?

This is the alternative!



Why Reproducible Research?

This is the alternative!



I dare you to try and reproduce my results!
(or check them for accuracy)

The Change Cascade

What if I need to make a change?

The Change Cascade

What if I need to make a change?

- Noticed an error
- Received new or additional data

The Change Cascade

What if I need to make a change?

- Noticed an error
- Received new or additional data

How many different places do I need to update?

The Change Cascade

What if I need to make a change?

- Noticed an error
- Received new or additional data

How many different places do I need to update?→ *Wasted Time*

The Change Cascade

What if I need to make a change?

- Noticed an error
- Received new or additional data

How many different places do I need to update?→ *Wasted Time*

What if we need to make a change in a year (or two)?

The Change Cascade

What if I need to make a change?

- Noticed an error
- Received new or additional data

How many different places do I need to update?→ *Wasted Time*

What if we need to make a change in a year (or two)?

What did I do again???

Elements of Reproducibility

What do we need to do to make analyses reproducible?

Elements of Reproducibility

What do we need to do to make analyses reproducible?

- Organization (where are my car keys?)

Elements of Reproducibility

What do we need to do to make analyses reproducible?

- Organization (where are my car keys?)
  - Standard folder structure and naming conventions
  - Code–Report pairing

Elements of Reproducibility

What do we need to do to make analyses reproducible?

- Organization (where are my car keys?)
  - Standard folder structure and naming conventions
  - Code–Report pairing
- Literate Programming
  - Well documented code
  - "Human readable"

Elements of Reproducibility

What do we need to do to make analyses reproducible?

- Organization (where are my car keys?)
  - Standard folder structure and naming conventions
  - Code–Report pairing
- Literate Programming
  - Well documented code
  - "Human readable"
- Version control
  - Tracked changes from project inception
  - software and package versions controlled

Section 2

The Vision

The Tools

We can use tools to help with organization and version control

The Tools

We can use tools to help with organization and version control

- Organization
  - LaTeX or Rmarkdown typesetting language + knitr (with R studio)
  - HTG R Package

The Tools

We can use tools to help with organization and version control

- Organization
  - LaTeX or Rmarkdown typesetting language + knitr (with R studio)
  - HTG R Package
- Version control
  - Git
  - Checkpoint

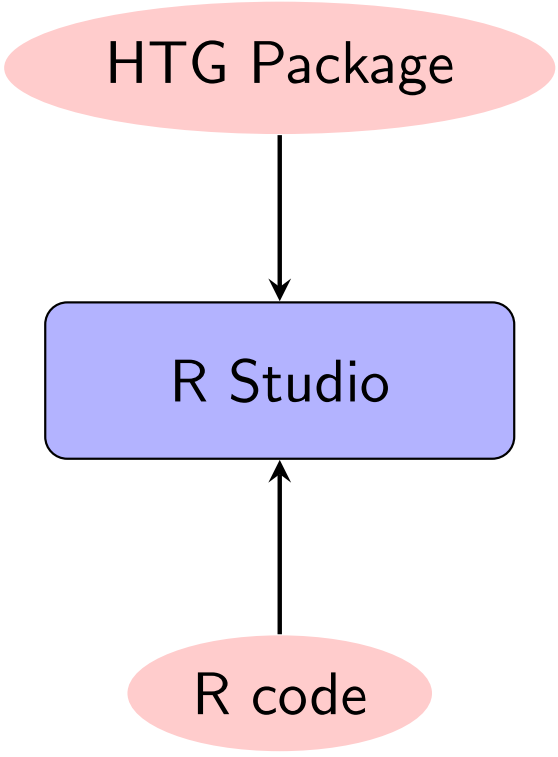
The Tools

We can use tools to help with organization and version control

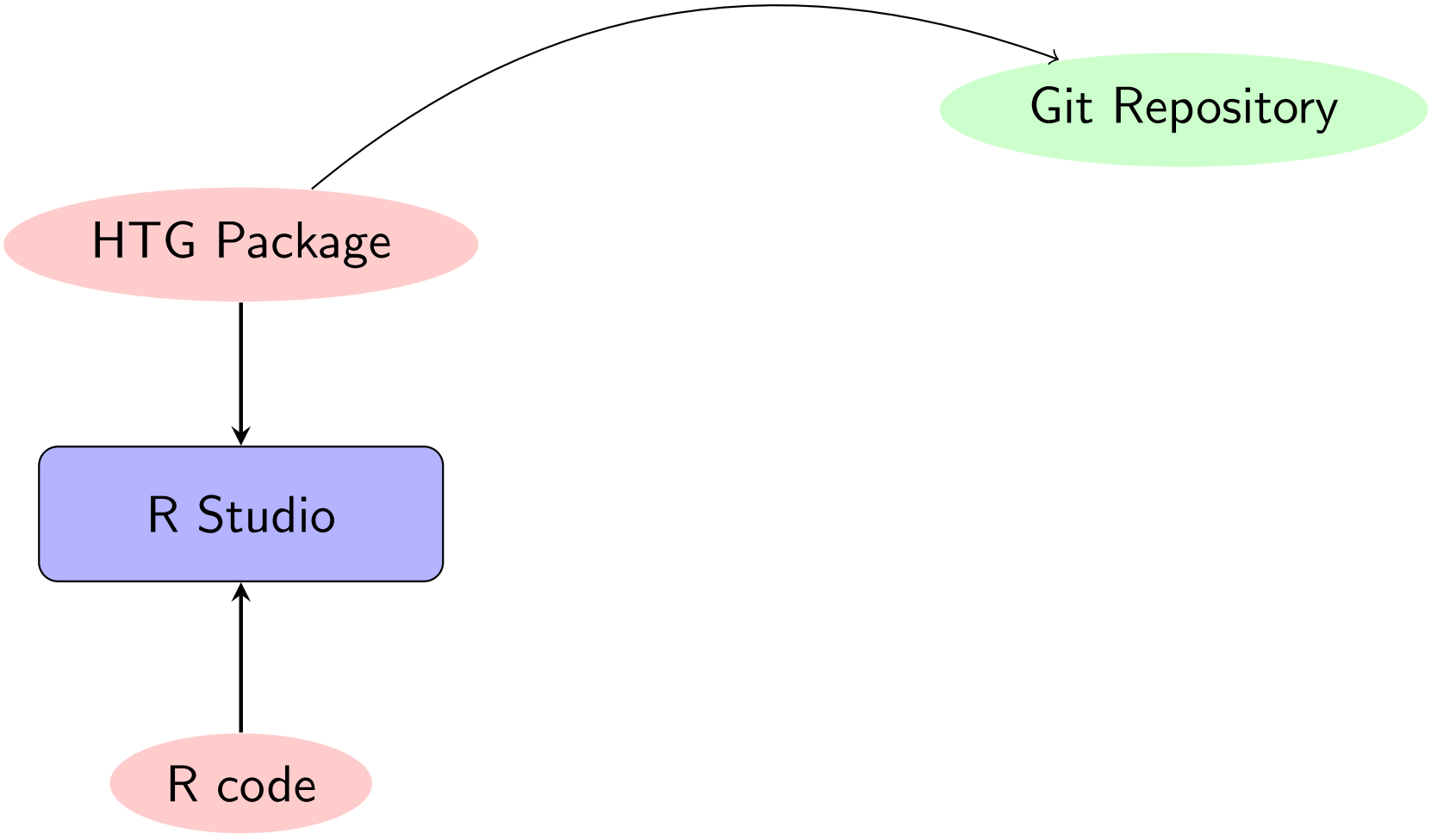
- Organization
  - LaTeX or Rmarkdown typesetting language + knitr (with R studio)
  - HTG R Package
- Version control
  - Git
  - Checkpoint

Use these tools together to create a seamless analysis $\rightarrow$ report pipeline.

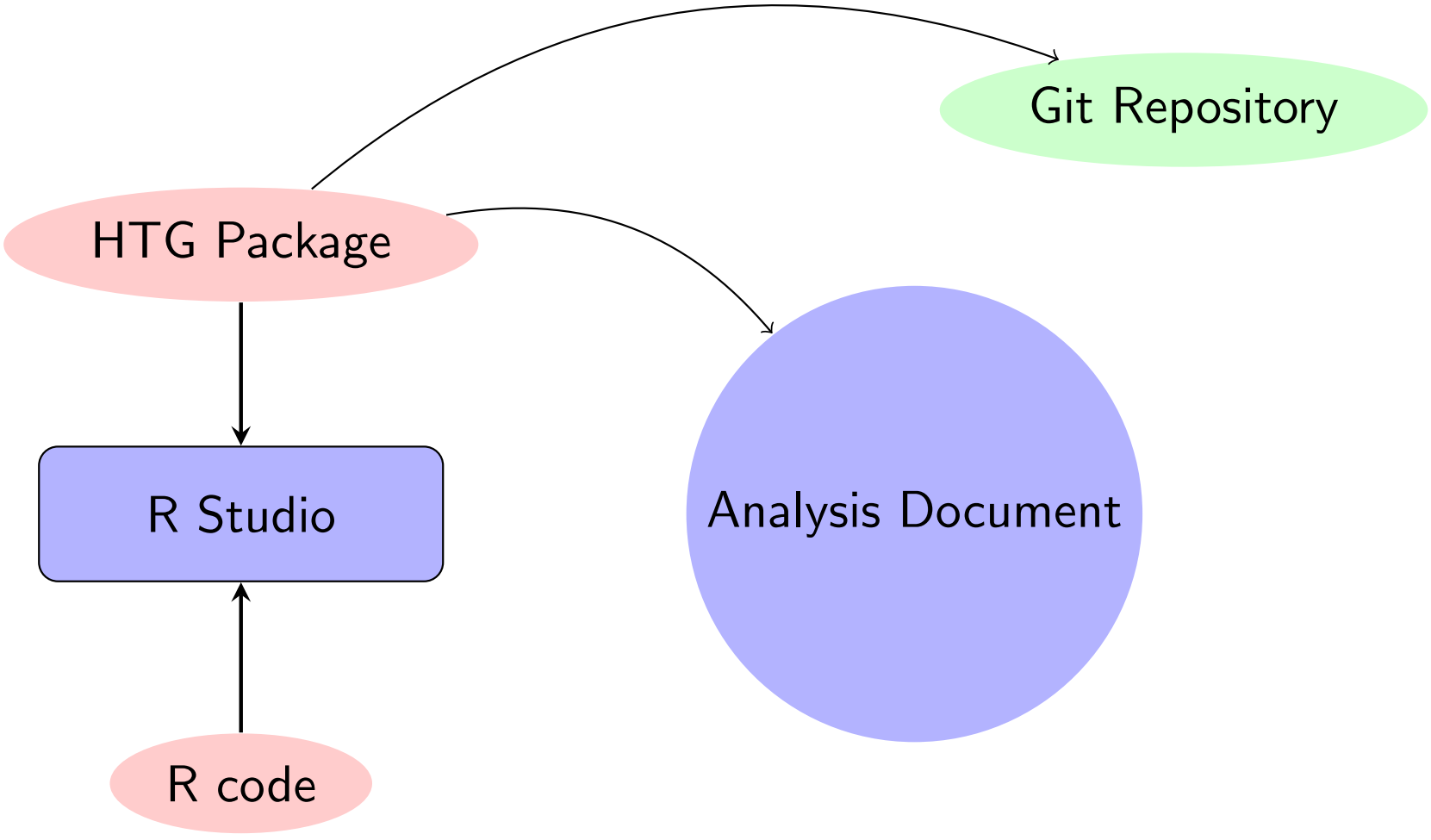
System Overview



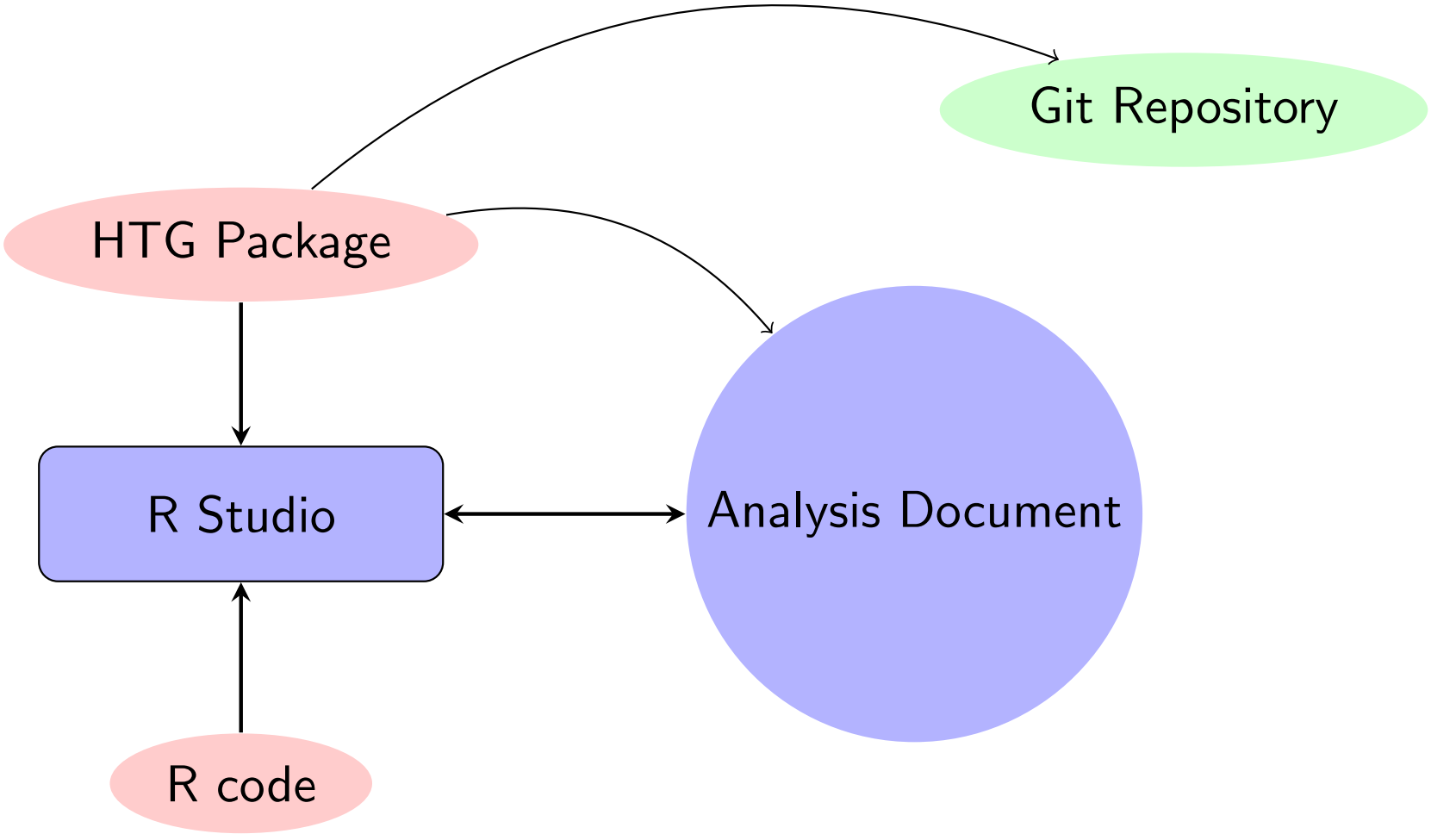
System Overview



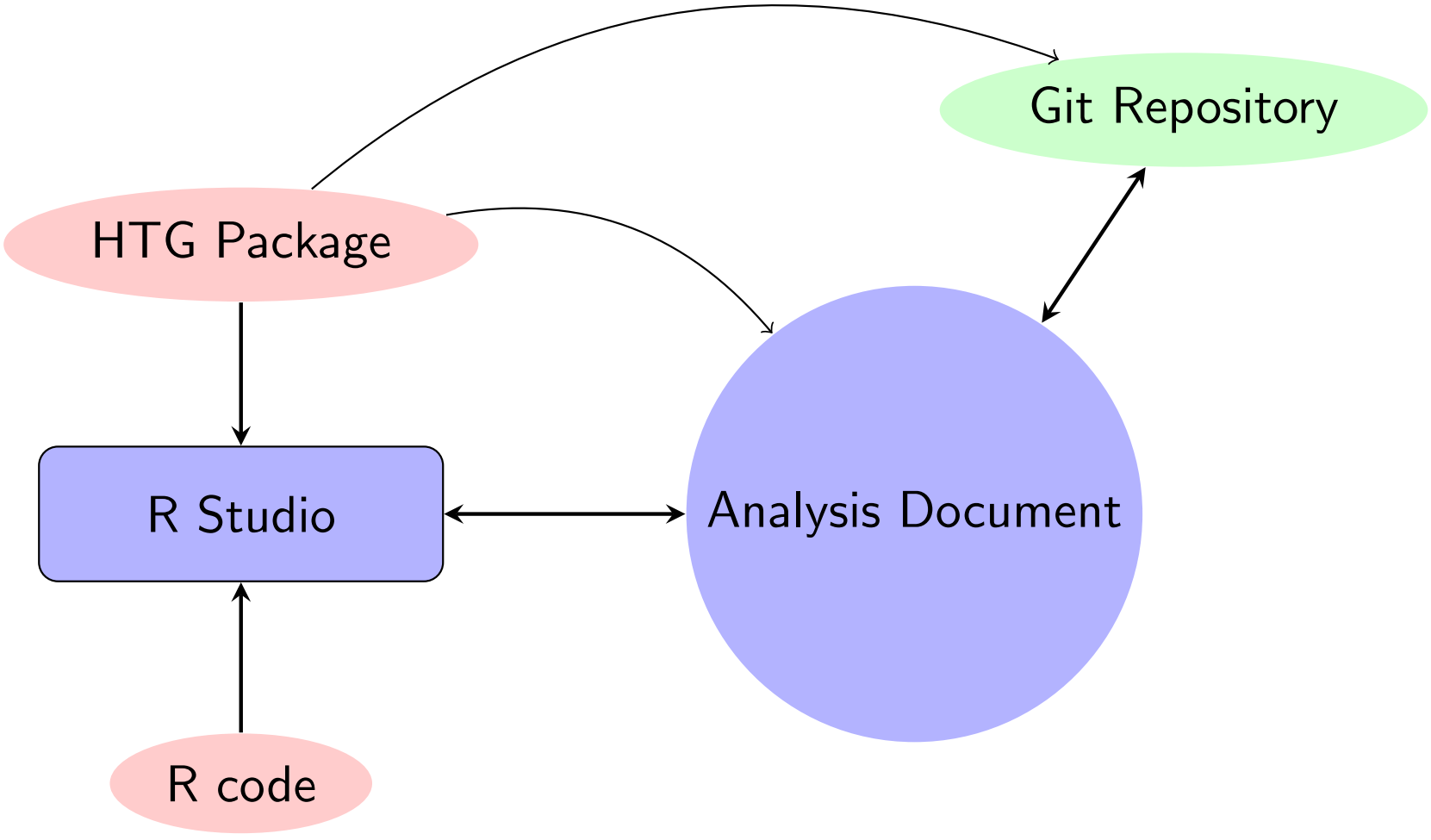
System Overview



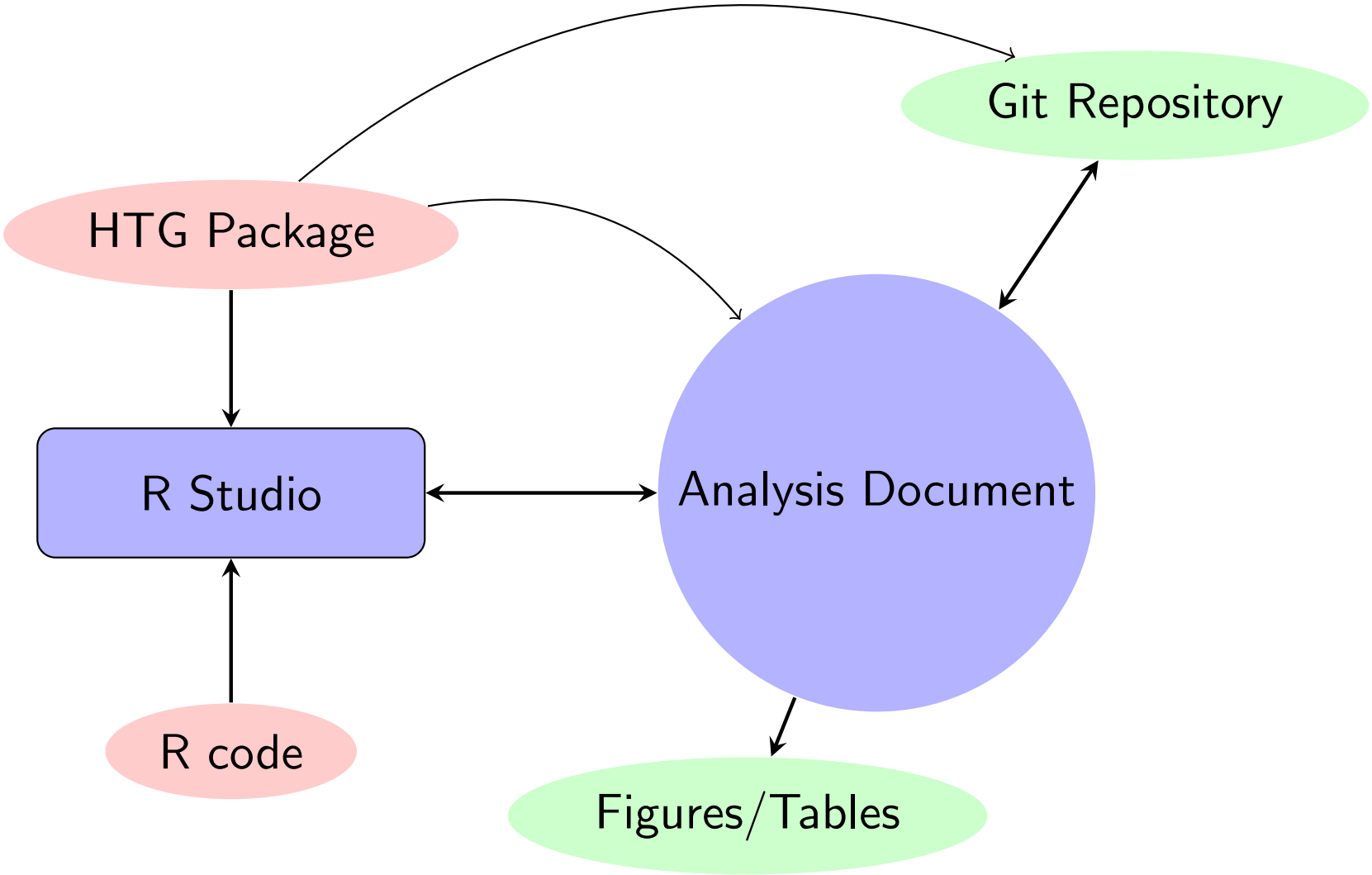
System Overview



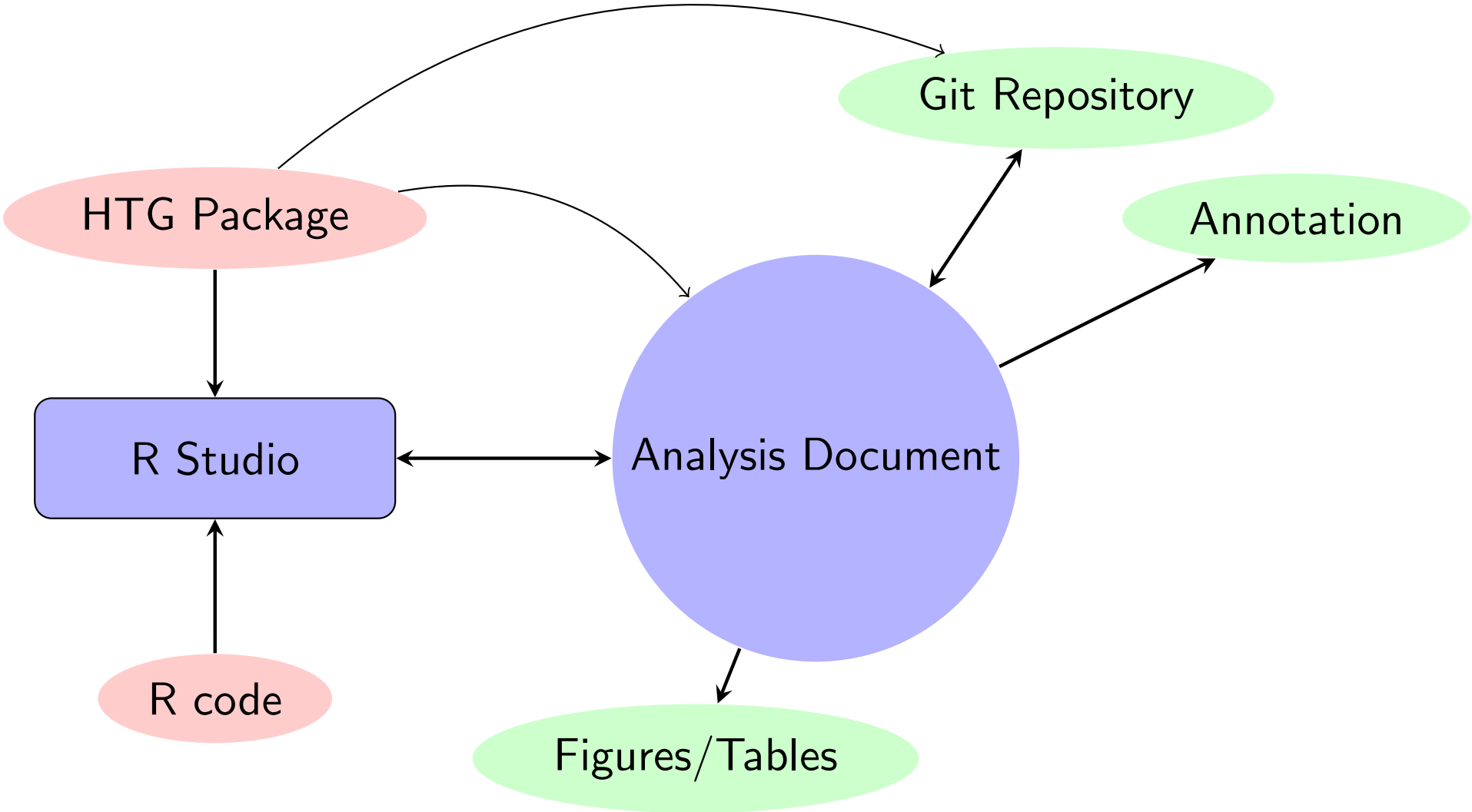
System Overview



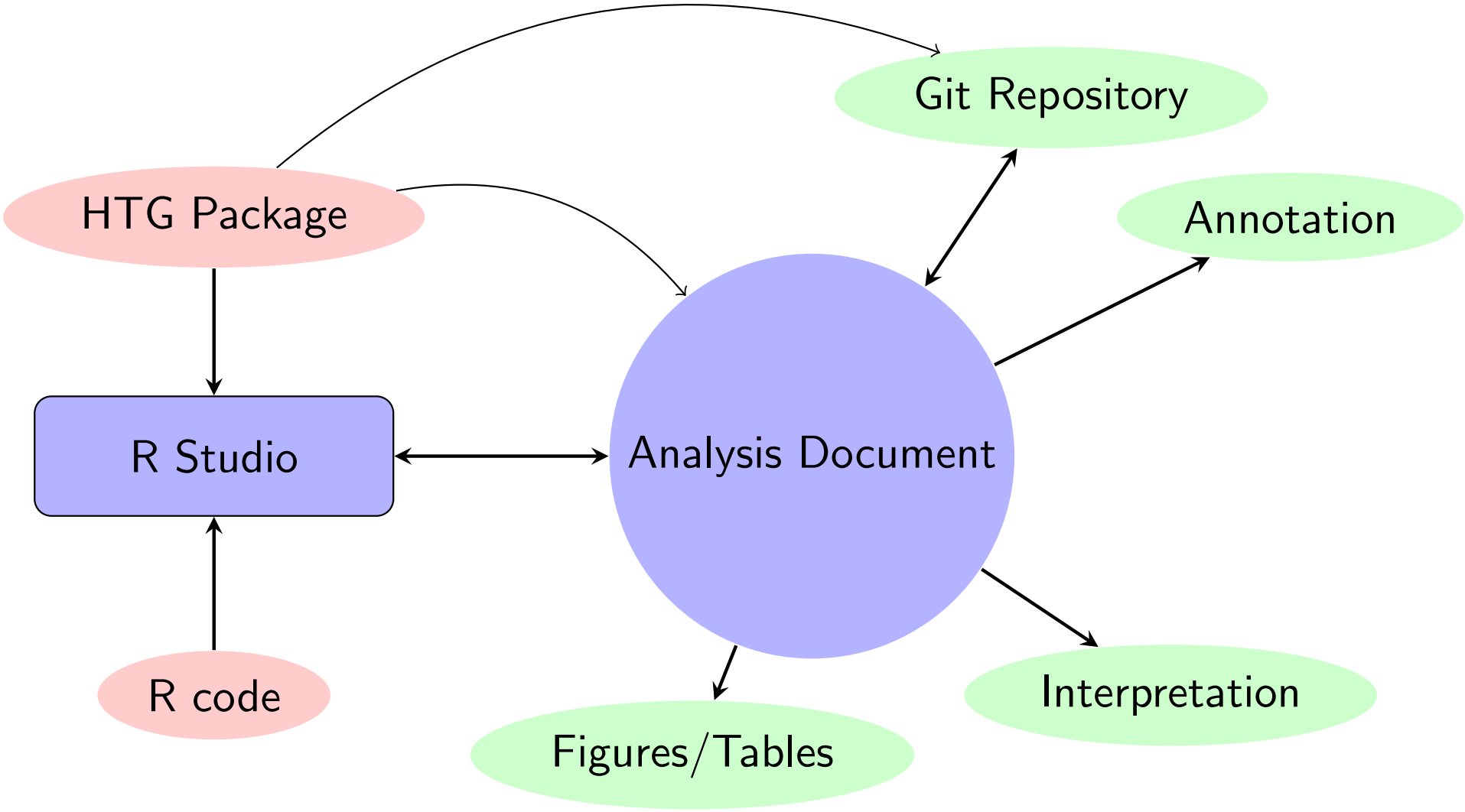
System Overview



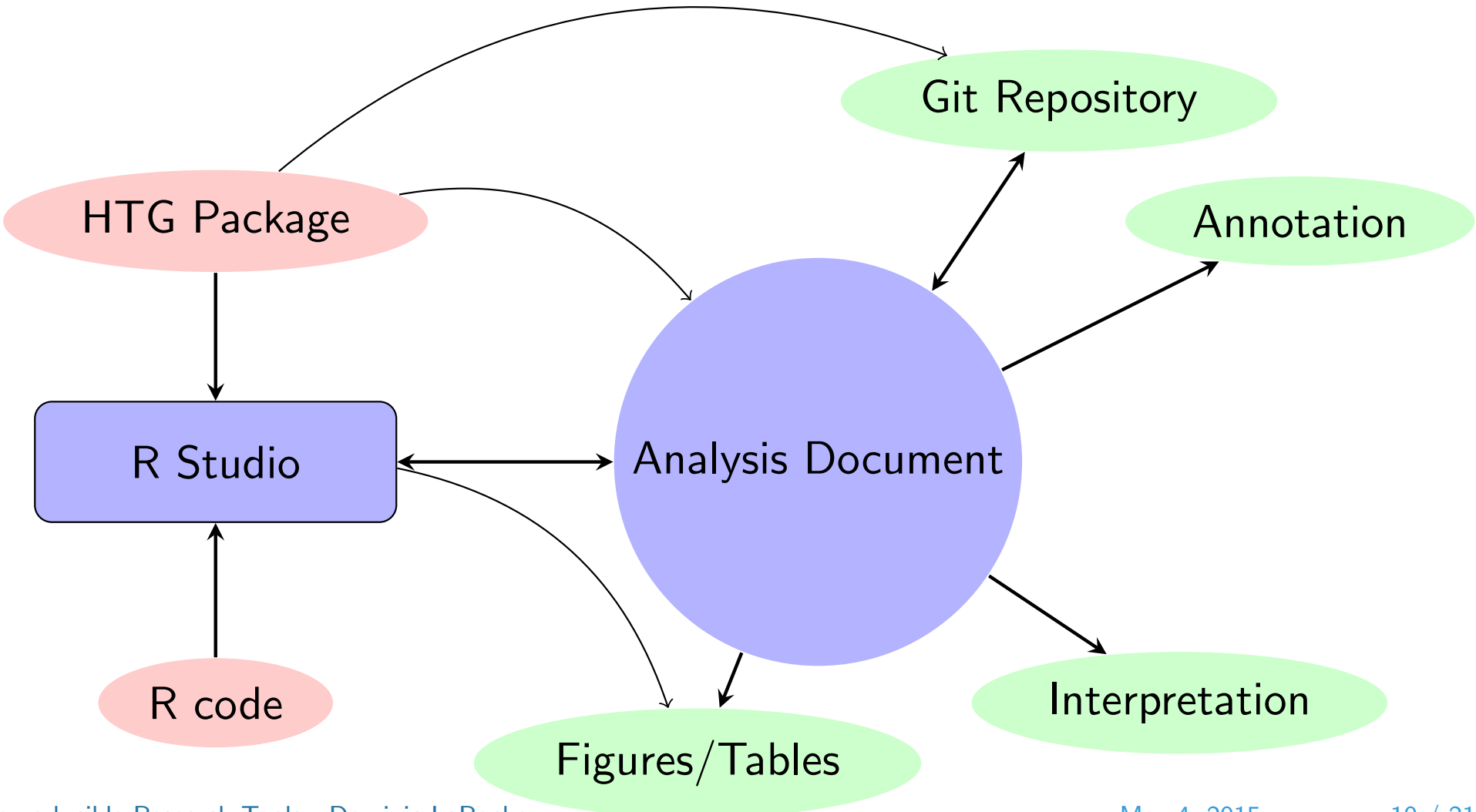
System Overview



System Overview



System Overview



Subsection 1

The Tools

R-Studio

R-Studio is a friendly code editor :

- Built in editor support for:
  - R code
  - C++
  - JavaScript
  - CSS
  - DCF
  - Markdown
  - HTML
  - Tex
  - Python
- It can also handle:
  - PERL
  - SAS ...etc
- Can enable emacs key-bindings

R-Studio

R-Studio combined with LaTeXor Rmarkdown can make:

- Reports
- Articles (formatted to most journals)
- Presentations
- Web pages and more...

R-Studio

R-Studio combined with LaTeXor Rmarkdown can make:

- Reports
- Articles (formatted to most journals)
- Presentations
- Web pages and more...

(This presentation was made with R-studio)

Git

Git is a version control system that also works with Rstudio

- Tracks changes to all documents in the folder or subfolders
  - This includes figures, pictures, emails, etc.

Git

Git is a version control system that also works with Rstudio

- Tracks changes to all documents in the folder or subfolders
  - This includes figures, pictures, emails, etc.
- Can retrieve any previous version of a file

Git

Git is a version control system that also works with Rstudio

- Tracks changes to all documents in the folder or subfolders
  - This includes figures, pictures, emails, etc.
- Can retrieve any previous version of a file
- No need to create multiple copies of a document

Git

Git is a version control system that also works with Rstudio

- Tracks changes to all documents in the folder or subfolders
  - This includes figures, pictures, emails, etc.
- Can retrieve any previous version of a file
- No need to create multiple copies of a document
  - Changes are tracked through commits

Git

Git is a version control system that also works with Rstudio

- Tracks changes to all documents in the folder or subfolders
  - This includes figures, pictures, emails, etc.
- Can retrieve any previous version of a file
- No need to create multiple copies of a document
  - Changes are tracked through commits
  - Each commit is attached to a (detailed) message about the changes/ status of the document

Git

Git is a version control system that also works with Rstudio

- Tracks changes to all documents in the folder or subfolders
  - This includes figures, pictures, emails, etc.
- Can retrieve any previous version of a file
- No need to create multiple copies of a document
  - Changes are tracked through commits
  - Each commit is attached to a (detailed) message about the changes/ status of the document
- Can make "clones" so that anyone can get the most current version of everything with a single command

Git

Git is a version control system that also works with Rstudio

- Tracks changes to all documents in the folder or subfolders
  - This includes figures, pictures, emails, etc.
- Can retrieve any previous version of a file
- No need to create multiple copies of a document
  - Changes are tracked through commits
  - Each commit is attached to a (detailed) message about the changes/ status of the document
- Can make "clones" so that anyone can get the most current version of everything with a single command
- Can specify items you don't want tracked

Git

Git is a version control system that also works with Rstudio

- Tracks changes to all documents in the folder or subfolders
  - This includes figures, pictures, emails, etc.
- Can retrieve any previous version of a file
- No need to create multiple copies of a document
  - Changes are tracked through commits
  - Each commit is attached to a (detailed) message about the changes/ status of the document
- Can make "clones" so that anyone can get the most current version of everything with a single command
- Can specify items you don't want tracked
- Can "Fork" a project into a new project

Checkpoint

Checkpoint is an R package to keep track of package versions

- Git doesn't track changes to the computing environment

Checkpoint

Checkpoint is an R package to keep track of package versions

- Git doesn't track changes to the computing environment
- Checkpoint creates a snapshot of all R packages used in the analysis

Checkpoint

Checkpoint is an R package to keep track of package versions

- Git doesn't track changes to the computing environment
- Checkpoint creates a snapshot of all R packages used in the analysis
- Package versions are maintained on a separate server (using a git style change tracker).

Checkpoint

Checkpoint is an R package to keep track of package versions

- Git doesn't track changes to the computing environment
- Checkpoint creates a snapshot of all R packages used in the analysis
- Package versions are maintained on a separate server (using a git style change tracker).
- Can access the working version of any CRAN package at any time
  - No need to worry that, *ahem*, someone is using a really old version of the package
  - Can keep your system up-to-date without worrying about breaking an analysis

HTG Private Package

The HTG package eases implementation of reproducible research

- MakeProject()
  - Single command creates version controlled project folder
  - Creates analysis/report templates with either Rmarkdown or LaTeX
  - Creates README file for the project to facilitate communication and collaboration
  - Creates Rproject file

HTG Private Package

The HTG package eases implementation of reproducible research

- MakeProject()
  - Single command creates version controlled project folder
  - Creates analysis/report templates with either Rmarkdown or LaTeX
  - Creates README file for the project to facilitate communication and collaboration
  - Creates Rproject file
- CloneProject()
  - Clones existing version controlled project to your computer

HTG Private Package

The HTG package eases implementation of reproducible research

- MakeProject()
  - Single command creates version controlled project folder
  - Creates analysis/report templates with either Rmarkdown or LaTeX
  - Creates README file for the project to facilitate communication and collaboration
  - Creates Rproject file
- CloneProject()
  - Clones existing version controlled project to your computer
- Can be a place to keep other standard functions

The Glue

R-studio makes it easy to implement these tools

- Weaves code and text together to make reports
  - Works with Rmarkdown or LaTeX
  - Rmarkdown lowers the bar for making reports

The Glue

R-studio makes it easy to implement these tools

- Weaves code and text together to make reports
  - Works with Rmarkdown or LaTeX
  - Rmarkdown lowers the bar for making reports
- Has easy interface with Git
  - Can see change history and associated hashes
  - has simple pull, commit and push cammands etc.

Outside Groups

How do we maintain reproducibility outside our group?

Outside Groups

How do we maintain reproducibility outside our group? Shiny applications

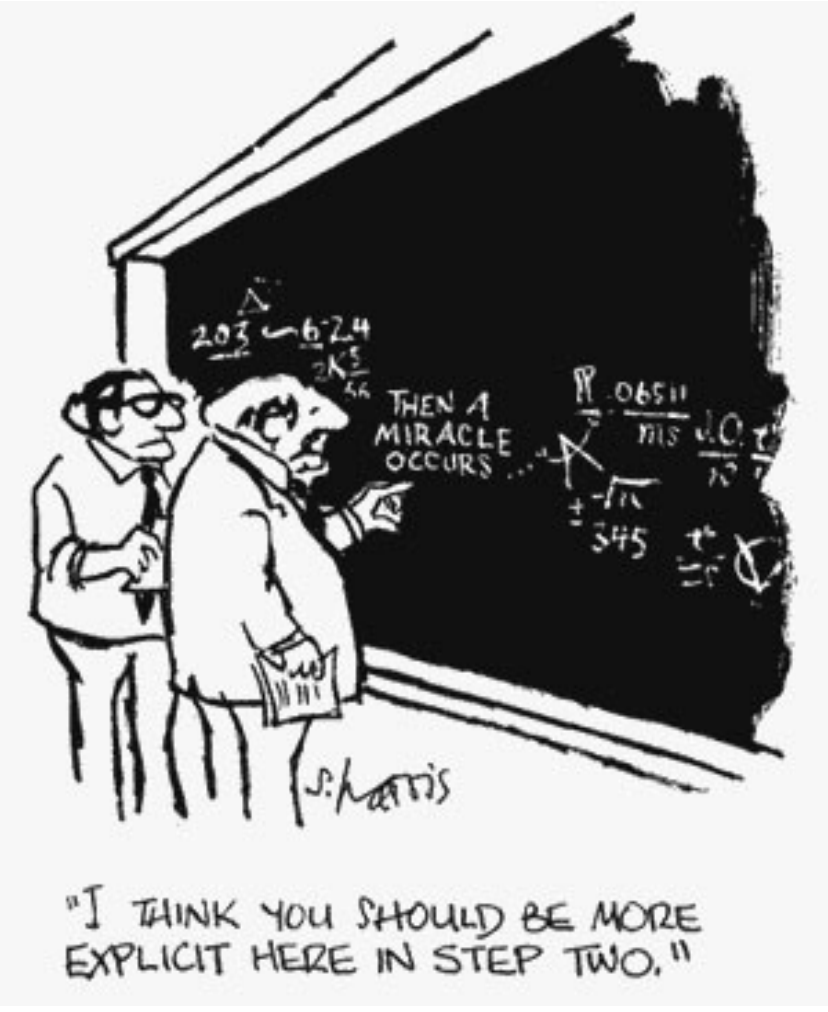
- Provide easy-to-use program to perform a specific task
- Eliminates use of other programs and click-through analyses
- Can incorporate documentation (exported code and report)
- Demo...

Section 3

Conclusions

Benefits

- Standardized file structure
- Access to all previous versions of everything
- HTG package eases set-up and saves time
- Anyone can pick up where you left off or find important figures and documents


THEN A MIRACLE OCCURS...
"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

Questions?