Script for Git Example

# Create a new project

**Kevin**

Scenario: I am starting a new project.

Steps:

* Create a new repo in GitHub (*Example Project*)
* Create a new project in RStudio (*Example Project*) that is connected to the repo

# Create a data analysis product

**Kevin**

Scenario: I was asked if I could make a quick chart with the provided data.

Steps:

* Create new R script called *datcon\_example.R*
* Copy the code below to the script
* Briefly walk through the code
* Run the code to show the plot

# Commit and push everything

**Kevin**

Scenario: The product was created, but I don’t want to just leave the project like that.

Steps:

* Commit those updates
* Push the updates

# Teammate updates the project

**Andrea**

Scenario: The stakeholder asked for some changes to the product, but Kevin is OOO. Andrea is asked to make those changes.

Steps:

* Open the *Git-Session-Jan-25-DatCon* project in RStudio
* Show that there is no script in your project (src/datacon\_example.R)
* *Pull* to get the script
* Open the script (*datcon\_example.R*)
* Run the script to show that the code is working
* In the ggplot chunk, fix the spelling error in the title line (“HHouseholds”) and change the fill color from *(“green”, “red”, “gray”)* to *(“blue”, “red”, “gray”)*.
* Run the script again to show the changes to the plot.
* Commit the changes to the script (any method you prefer)
* Push those changes

# Return to the project to continue working on it

**Kevin**

Scenario: I return the following week and see the old email about the updates to I go into the project to try and update.

Steps:

* Discuss how I should do a pull first
* Instead, open the script and change the color to (“orange”, “blue”, “gray”). No change to the title spelling
* Save, commit, and try to push the changes.
* See there is a conflict and explore
* Fix the conflict and commit and push again.