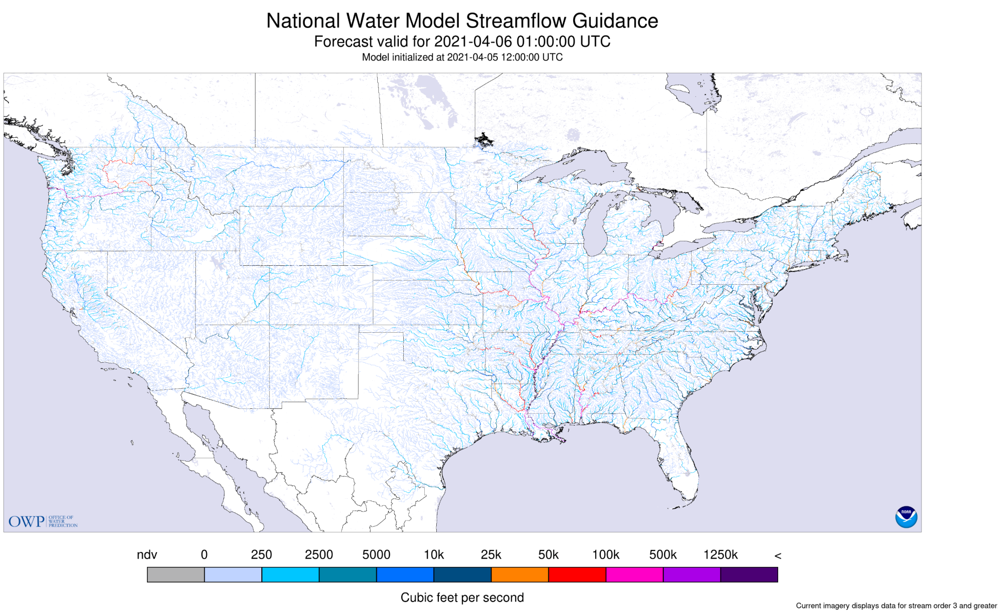
NWM Shiny application

Geography 176C

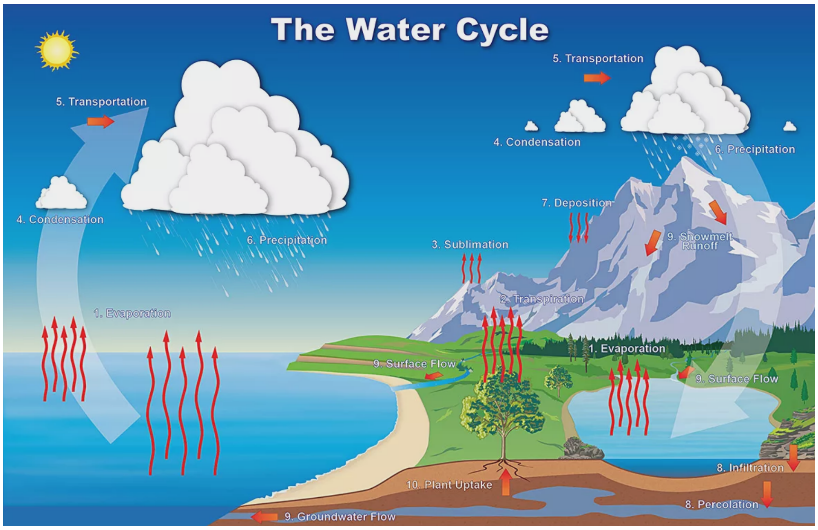
Angus Watters

R Shiny streamflow application



The National Water Model (NWM)

- The National Water Model (NWM) project uses USGS stream gauges to simulate the water cycle processes using mathematical representations for different processes to see how the processes fit together.



Problem

There is **NOT** an easy way to view simultaneously view catchments + stream flow data

–

Solution

- Create a ***R Shiny dashboard*** that allows the user to explore all watershed catchments in the US and get on demand streamflow data

Research questions

–

1. What is the relative impact of certain geographic factors on stream flows?

—
-
Proximity to a dam, population density, land cover
—

2. Forecast stream flow rates using precipitation and water withdrawals in a multilayer perceptron Machine learning algorithm

---

Data sources

–

- Project and data retrieval will use these **R packages**:
  - `tidyr`
  - `dplyer`
  - `sf`
  - `leaflet`
  - `dataRetrieval`
  - `nhdplusTools`
  - `nwmTools`
  - `shiny`

count: false

Find flowline from map click

.panel1-nldi-auto[

```r
library(dataRetrieval)

# R Client for the Network Linked Data Index
flowline <- findNLDI(location = c(-94.64, 31.09))  #<<
```

]

.panel2-nldi-auto[]

count: false

Find flowline from map click

.panel1-nldi-auto[

```
library(dataRetrieval)

# R Client for the Network Linked Data Index
flowline <- findNLDI(location = c(-94.64, 31.09))
flowline
```

]

.panel2-nldi-auto[

```
Simple feature collection with 1 feature and 3 fields
geometry type:  LINESTRING
dimension:      XY
bbox:           xmin: -94.64845 ymin: 31.0838 xmax: -94.62997 ymax: 31.09915
geographic CRS: WGS 84
# A tibble: 1 x 4
  sourceName   identifier comid                                         geometry
  <chr>        <chr>      <chr>                                 <LINESTRING [°]>
1 NHDPlus com~ 101        101   (-94.64845 31.09915, -94.64803 31.09871, -94.64~
```

]

---

count: false

Find catchment polygons from COMID

.panel1-nhd-auto[

```
library(nhdplusTools)  #<<
```

]

.panel2-nhd-auto[]

count: false

Find catchment polygons from COMID

.panel1-nhd-auto[

```
library(nhdplusTools)

# Get catchment polygons from National Hydrography Dataset V2 Subsets
catchment <- get_nhdplus(comid = flowline$comid,
                         realization = "catchment")
catchment  #<<
```

]

.panel2-nhd-auto[

```
Simple feature collection with 1 feature and 7 fields
geometry type:  POLYGON
dimension:      XY
bbox:           xmin: -94.6561 ymin: 31.0837 xmax: -94.6285 ymax: 31.108
geographic CRS: WGS 84
# A tibble: 1 x 8
  id               gridcode featureid sourcefc   areasqkm shape_length shape_area
* <chr>               <int>     <int> <chr>         <dbl>        <dbl>      <dbl>
1 catchmentsp.18~  1605970       101 NHDFlowli~     4.57        0.100   0.000432
# ... with 1 more variable: geometry <POLYGON [°]>
```

]

count: false

View catchments in leaflet

.panel1-interactive-auto[

```r
library(leaflet)  #<<
```

]

.panel2-interactive-auto[]

count: false

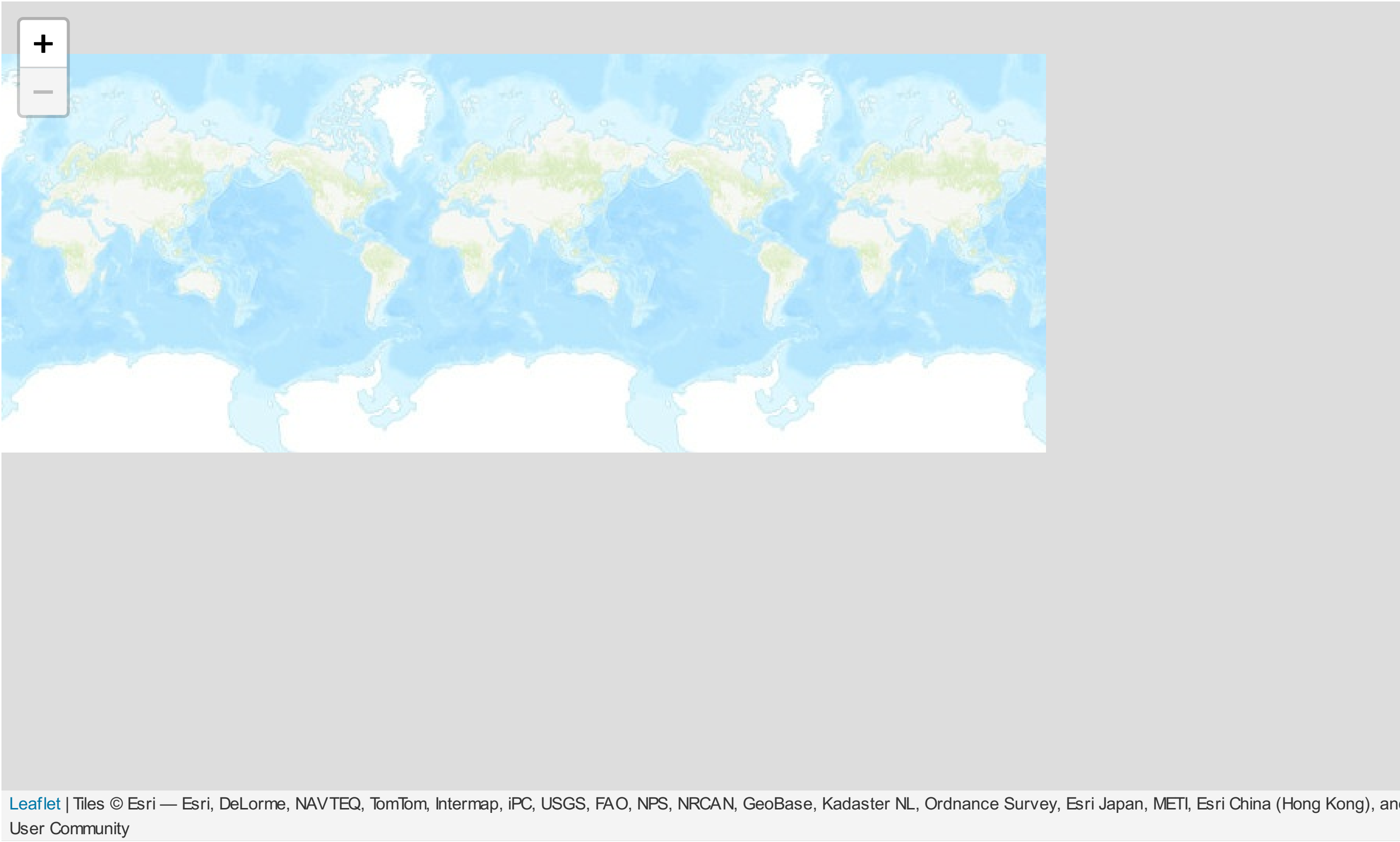
View catchments in leaflet

.panel1-interactive-auto[

```r
library(leaflet)

leaflet() %>%
  addProviderTiles(providers$Esri.WorldTopoMap)   #<<
```

]

.panel2-interactive-auto[



]

count: false

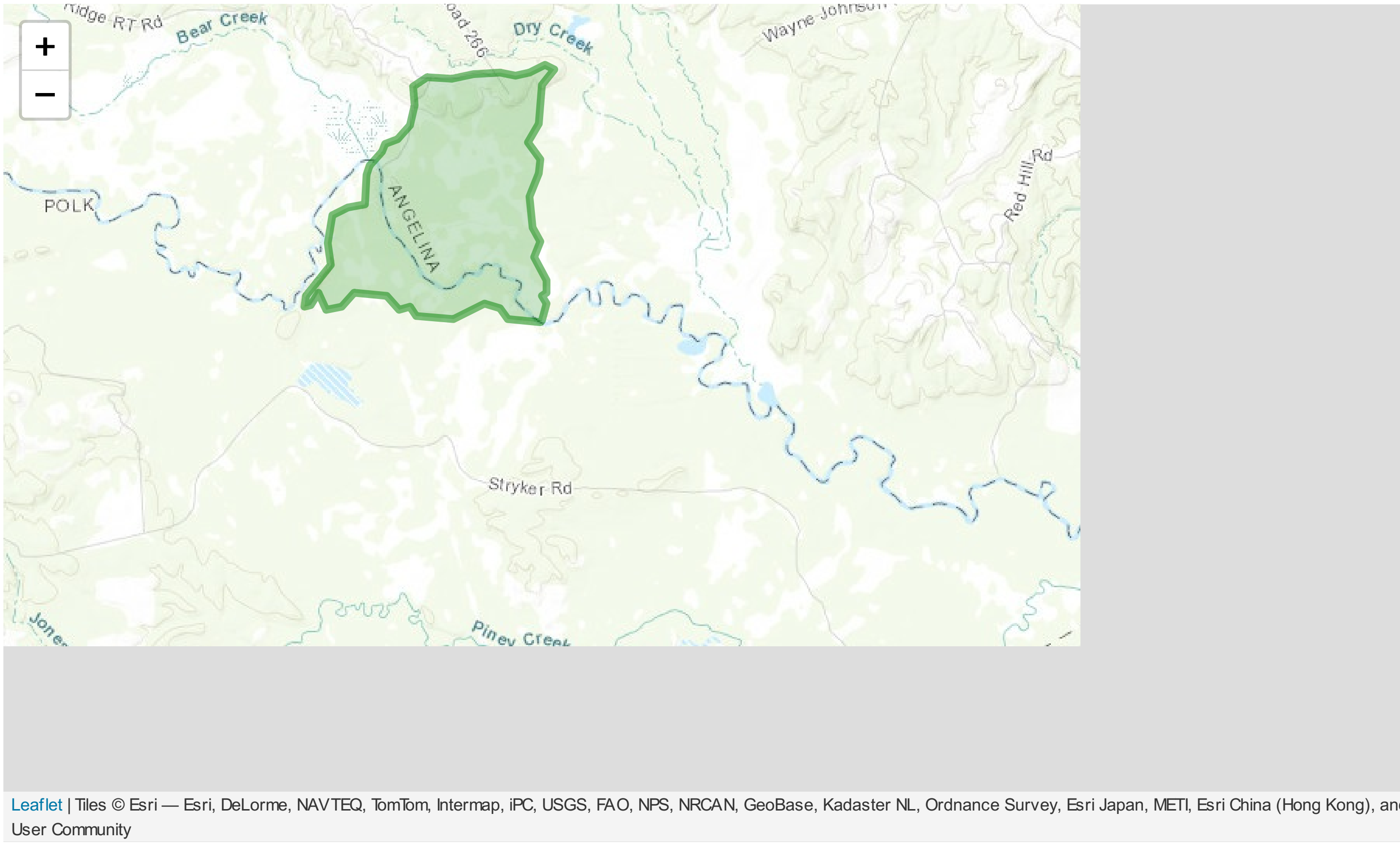
View catchments in leaflet

.panel1-interactive-auto[

```r
library(leaflet)

leaflet() %>%
  addProviderTiles(providers$Esri.WorldTopoMap) %>%
  setView(-94.64, 31.09, 13) %>%
  addPolygons(data = catchment,   #<<
              color = "green")    #<<
```

]

.panel2-interactive-auto[



]

count: false

NWM data using nwmTools package

.panel1-nwm-auto[

```
library(nwmTools)   #<<
```

]

.panel2-nwm-auto[]

count: false

NWM data using nwmTools package

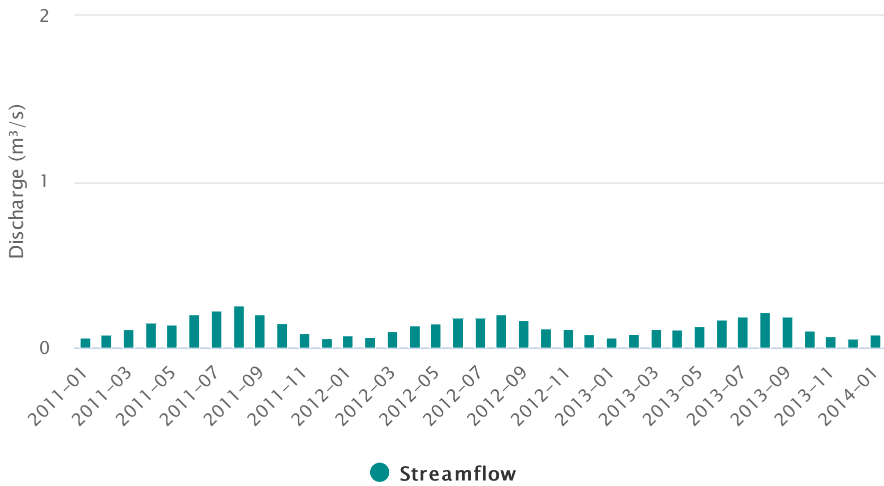
.panel1-nwm-auto[

```
library(nwmTools)

# Retrieve flow data from NWM via COMID
```

```
# nwm <- nwmTools::readNWMdata(comid = 101)  # nwm <- nwmTools::readNWMdata(comid = 101)
graph_NWM(nwm)  #<<
```

]

.panel2-nwm-auto[



]

---

count: false

Climate data using climateR package

.panel1-climate-auto[

```
library(climateR)  #<<
```

]

.panel2-climate-auto[]

count: false

Climate data using climateR package

.panel1-climate-auto[

```
library(climateR)
library(sf)
library(highcharter)  #<<
```

]

.panel2-climate-auto[]

count: false

Climate data using climateR package

.panel1-climate-auto[

```
library(climateR)
library(sf)
library(highcharter)

# Retrieve climate data for the catchment using centroid point
pt <- st_centroid(flowline)

# precip <- getTerraClim(AOI = pt, param = "prcp", startDate = "2011-01-01", endDate = "2014-01-01")  #
```

]

.panel2-climate-auto[]

count: false

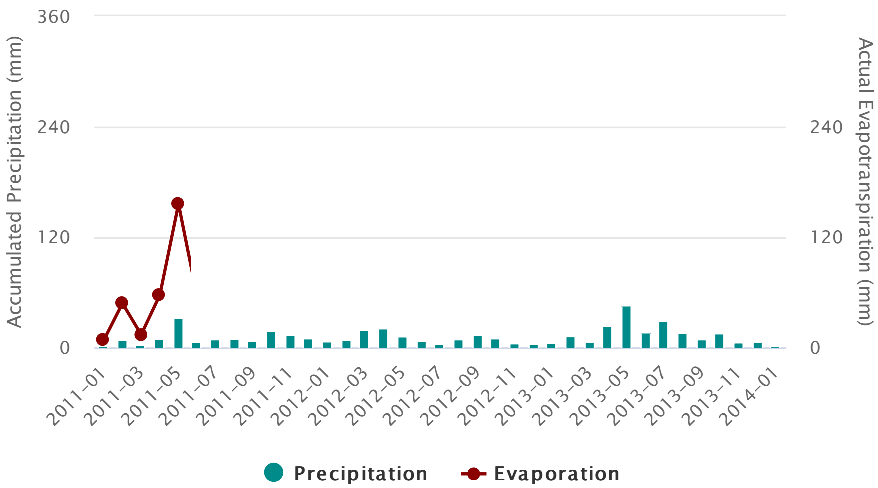
Climate data using climateR package

.panel1-climate-auto[

```
library(climateR)
library(sf)
library(highcharter)

# Retrieve climate data for the catchment using centroid point
pt <- st_centroid(flowline)

# precip <- getTerraClim(AOI = pt, param = "prcp", startDate = "2011-01-01", endDate = "2014-01-01")  #
# evap <- getTerraClim(AOI = pt, param = "aet", startDate = "2011-01-01", endDate = "2014-01-01")  # ev

graph_climateR(precip, evap)  #<<
```

]

.panel2-climate-auto[


360
240
120
0
Accumulated Precipitation (mm)
Actual Evapotranspiration (mm)
2011-01
2011-03
2011-05
2011-07
2011-09
2011-11
2012-01
2012-03
2012-05
2012-07
2012-09
2012-11
2013-01
2013-03
2013-05
2013-07
2013-09
2013-11
2014-01
Precipitation
Evaporation

]