R-ArcGIS Bridge Workshop Cheat Sheet

If you find yourself forgetting any R functions that you've learned during the Integrating R with ArcGIS workshop, use this guide as a reference.

dplyr functions

filter() – Subset rows from a data frame. Similar in function to base R subsetting.

```
filter(crime_df, Arsons > 3, Thefts > 10)
```

arrange() – Sort rows in a data frame based on a set of column names. Can sort by multiple different columns.

```
arrange(crime_df, Arsons, Assaults)
```

select() – Select specified columns (or variables) from a data frame.

```
select(crime_df, AREA_S_CD, Equity_Score)
```

summarize() – Summarize values from a data frame given a function, and collapse results to a single row (unless data are grouped).

```
summarize(crime_df,
          mean_fire = mean(Fire.Vehicle.Incidents, na.rm = TRUE))
```

summarize_each() – Summarize values from a data frame given multiple functions.

```
summarize_each(crime_df, c('mean', 'sd'), Equity_Score)
```

%>% (Forward-pipe operator) – Allows you to pipe a value forward into an expression or function call, e.g., `f(x, y)` become `x %>% f(y)`.

```
crime_df %>%
  filter(Assaults == 0) %>%
  select(Equity_Score, Thefts) %>%
  arrange(Thefts)
```

group_by() – Group a data frame given a variable (or list of variables).Groups will be used when you apply functions to this data frame.

```
arson_groups = group_by(crime_df, Arsons)
summarize(arson_groups, mean_fire = mean(Fire.Vehicle.Incidents, na.rm =
TRUE))
```

Adding an underscore to the end of any of these functions (e.g., `arrange_()`) to be able to pass parameters as lists (or more so, vectors).

```
sort_fields = c('Arsons', 'Thefts')
arrange_(crime_df, .dots = sort_fields)
```

arcgisbinding functions

arc.check_product() – Run after loading arcgisbinding package to authenticate ArcGIS license.

```
arc.check_product()
```

arc.open() – Connect to a feature class, table, or raster dataset to R from a specified path.

```
input_fc = arc.open('data.gdb/features')
```

arc.select() – Load a feature class into an ArcGIS data frame.

```
arc.select(tor_crime, fields = c('Neighbourhood', 'Arsons'))
```

arc.raster() – Load a feature class into an ArcGIS data frame.

```
arc.raster(arc.open('data.gdb/raster'))
```

arc.data2sp() – Create sp object for an ArcGIS data frame.

```
tor_crime_sp = arc.data2sp(tor_crime_df)
```

arc.data2sf() – Create sf object for an ArcGIS data frame.

```
tor_crime_sf = arc.data2sf(tor_crime_df)
```

as.raster() – Create a RasterLayer or RasterBrick object from an arc.raster data object:

```
as.raster(arc_raster_object)
```

arc.write() – Write out a feature class, table, or raster data object to a specified path.

```
arc.write('data.gdb/new_features', tor_crime_sp)
```

R-ArcGIS Script Tools

tool_exec(in_params, out_params) – Execute an R-ArcGIS script tool.

in_params – A list of parameters provided as inputs for the R script.

out_params – A list of parameters required when outputting data to ArcGIS.

return(out_params) – return output parameter list when script is completed.

arc.progess_label('Processing…') – set the label of the geoprocessing dialog progress indicator

arc.progess_pos(50) – set the position of the geoprocessing progress indicator (as a percentage)

message('<text>') – print messages in the geoprocessing dialog

warning('<text>') – print warnings in the geoprocessing dialog

stop('<text>') – stop execution and print an error in the geoprocessing dialog

RStudio Shortcuts

Insert the assignment operator `<-` : ALT+-

Insert the forward-pipe operator `%>%` : CTRL+SHIFT+M

Execute current line or selected code : CTRL+Enter

Run all code in current R script : CTRL+ALT+R

http://hed.esri.ca