Q'S ÜBER HELPFUL ONE LINERS

bash

Create and export a function in `.bashrc` which is just like `Rscript` except that `--no-echo` is removed.

```
function Rscript2() { R --no-restore --file="$1"; }
export -f Rscript2
```

Alias to replicate, very simply, the behavior of `dos2unix` to correct line endings of files created on Windows.

```
alias tounix='sed -i.bak "s/\r$//"'
```

Get rid of spaces in all file names in a directory, with a for loop.

```
for oldname in *; do newname=`echo $oldname | sed -e 's/ //g'`; mv "$oldname" "$newname"; done
```

git

Show the names of files that were modified in the last 10 commits

```
git diff --name-only HEAD~10 HEAD
```

sed

Replace all occurrences of `string1` with `string2` in all `.md` files in current directory

```
find . -name '*.md' -exec sed -i -e 's/string1/string2/g' {} \;
```

note: If there are forward slashes in the string to be replaced, you can use `|` as the separator.

R data.table

Using `purrr`, read all CSV files from a directory into memory, naming them after the base filename without extension.

```
walk(list.files('path/to/files', pattern = '*.csv', full.names = TRUE),
     ~ assign(gsub('\\.csv', '', basename(.)), fread(.), envir = .GlobalEnv))
```

note: Add additional arguments to ***`fread`*** *as needed.*

Do a left join keeping all rows in `df1`. This really only works for small numbers of columns.

```
df1[df2, on = .(foo,bar), `:=` (baz = i.baz, floop = i.floop)]
```

This is the same as above but for more columns.

```
DF1[DF2, on = .(date, id), names(DF2)[3:4] := mget(paste0("i.", names(DF2)[3:4]))]
```

Or if the vector of column names is created beforehand.

```
cols <- names(DF2)[3:4]
DF1[DF2, on = .(date, id), (cols) := mget(paste0("i.", cols))]
```

Fastest way to filter rows conditionally *within* each group. In this example we keep the row(s) with highest value of `v` in each group defined by `grp`. Use `which.max()` if you only want one row.

```
dt[dt[, .I[v == max(v)], by = grp]$V1]
```

One-hot encoding of a column, must have a unique ID column too.

```
y <- dcast(dat[, .(ID, factor_col)], ID ~ factor_col, length)
y[, ID := NULL]
```

Rmarkdown/knitr

Header to get the date in a good format:

```
date: "`r format(Sys.time(), '%B %d, %Y')`"
```

Alternative:

```
date: "`r format(Sys.time(), '%d %B %Y')`"
```

Figures side by side:

```
{r, fig.show="hold", out.width="50%"}
```

Show or hide code in an individual chunk:

```
{r, class.source = 'fold-show'}
```

(`fold-hide` to hide the code instead).

Windows command prompt

Symbolic link to a directory (run as admin). Replace the `/d` with `/J` to get a directory junction.

```
mklink /d data C:\stuff\foo
```

In R, run as admin, create a symlink in this way. Note in Windows the link comes first, in R and Linux shell the link comes last.

```
file.symlink(from = 'C:/Users/qdread/onedrive_usda/ars_projects/xxx', to = 'data')
```

Regex

In R, get all numbers between two specific character strings without actually getting the characters. The `?<=` within the initial parens means "find text preceded by ..." and `?=` means "find text followed by ..."

```
str_extract_all(x, '(?<=foo)([0-9]+)(?=bar)')
```

gt

Some options for a nice grouped table

```
tab_options(
    row_group.background.color = "#D4EBF2",
    row_group.font.weight = 'bold',
    column_labels.font.weight = 'bold'
  )
```

tidyverse

Pivoting by multiple column sets, using a name pattern (if the name is separated by an `_` as `grp2_metric`)

```
dat %>% pivot_longer(-grp1, names_to = c('grp2', '.value'), names_pattern = '(.+)_(.+)')
```

One-hot encoding

```
tibble %>%
    mutate(n = 1) %>%
    pivot_wider(names_from = column_to_encode, values_from = n, values_fill = 0)
```

ggplot2

Semitransparent median bar

```
stat_summary(
    fun = median, geom = "point",
    shape = 95, size = 50, alpha = 0.5
  )
```