PLOT_HUD_DESIGN

January 30, 2026 at 04:53 PM

Plot HUD Design Proposal

Date: 2026-01-30 **Status**: Proposal **Goal**: Unify plot management with existing HUD system for consistent UX

Current State

Existing HUD System (Consistent UX)

The HUD functions provide workspace visibility with consistent patterns:

HUD	Command	Key	Opens In
Memory	`:RMemoryHUD`	`<LocalLeader>m`	Vim split
Data Frames	`:RDataFrameHUD`	`<LocalLeader>e`	Vim split
Packages	`:RPackageHUD`	`<LocalLeader>z`	Vim split
Environment	`:REnvironmentHUD`	`<LocalLeader>x`	Vim split
Options	`:ROptionsHUD`	`<LocalLeader>a`	Vim split
Dashboard	`:RHUDDashboard`	`<LocalLeader>0`	5 tabs

Consistent UX patterns:

- Open in Vim splits or tabs
- Tabulated data display (Tabularize integration)
- q to close, / to search
- `<LocalLeader>` + single key for quick access
- Read-only buffers with viewer settings

Current Plot System (Inconsistent)

Feature	Implementation	Issue
Display	Kitty pane (external)	Different from HUDs
Watcher	Polling mechanism	Unique to plots
Navigation	R functions	Not Vim-native
Gallery	Vim buffer	Different styling
Key mappings	`<LocalLeader>p*` family	Inconsistent with HUD pattern

Proposed Design

Plot HUD Buffer
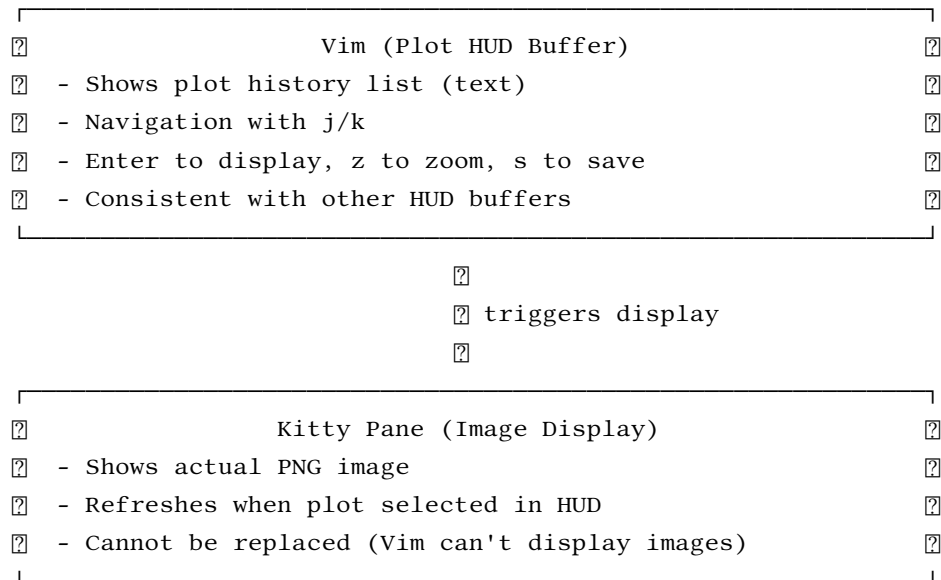
A new `:RPlotHUD` command (mapped to `<LocalLeader>P`) opens a Vim buffer styled like other HUDs:

```
Plot History                                            [HUD]
============================================================
Press Enter to display, z to zoom PDF, s to save, q to close

  #  Name                Created              Code
--------------------------------------------------------------------
> 1  scatter_mpg         2026-01-30 10:30     zzplot(mtcars$wt, mtcars$mpg)
  2  histogram           2026-01-30 10:35     zzplot(hist(rnorm(1000)))
  3  regression          2026-01-30 10:40     zzggplot(p + geom_smooth())


Total: 3 plots | Current: 1
```

Architecture

```
┌──────────────────────────────────────────────────────────┐
│                    Vim (Plot HUD Buffer)                 │
│  - Shows plot history list (text)                        │
│  - Navigation with j/k                                   │
│  - Enter to display, z to zoom, s to save                │
│  - Consistent with other HUD buffers                     │
└──────────────────────────────────────────────────────────┘
                        │
                        │ triggers display
                        │
┌──────────────────────────────────────────────────────────┐
│                  Kitty Pane (Image Display)              │
│  - Shows actual PNG image                                │
│  - Refreshes when plot selected in HUD                   │
│  - Cannot be replaced (Vim can't display images)         │
└──────────────────────────────────────────────────────────┘
```

Key Bindings (in Plot HUD buffer)

Key	Action	Consistent With
j/k	Move cursor	Standard Vim
Enter	Display selected plot in kitty pane	HUD pattern
z	Zoom - open PDF in Preview	Plot-specific
s	Save plot (prompt for filename)	Plot-specific
d	Delete plot from history	Plot-specific
q	Close HUD buffer	All HUDs
Esc	Close HUD buffer	All HUDs

Key	Action	Consistent With
/	Search	All HUDs
1-9	Quick select by number	Gallery pattern

Dashboard Integration

The HUD Dashboard (`<LocalLeader>0`) would include Plot HUD as 6th tab:

```
" Tab 6: Plot History
call s:CreateHUDTab('Plots', 'plots', function('s:GeneratePlotHUD'),
    \ l:source_file, l:r_terminal_id)
```

Commands

Command	Key	Description
:RPlotHUD	<LocalLeader>P	Open Plot HUD buffer
:RPlotZoom	(from HUD: z)	Open current PDF
:RPlotSave	(from HUD: s)	Save current plot

Implementation Functions

```
" Main HUD function - consistent with other HUDs
function! s:RPlotHUD() abort
    " Read .plots/history/index.json
    " Format as tabulated list
    " Open in split with viewer settings
    " Set up buffer-local mappings
endfunction

" Generate plot list for HUD tab
function! s:GeneratePlotHUD() abort
    " Called by CreateHUDTab for dashboard
    " Returns formatted plot history
endfunction

" Buffer-local action handlers
function! s:PlotHUDSelect() abort
    " Get plot under cursor, display in kitty pane
endfunction

function! s:PlotHUDZoom() abort
    " Open PDF of plot under cursor
endfunction

function! s:PlotHUDSave() abort
```

```
    " Save plot under cursor (prompt for filename)
endfunction
```

Benefits

1. **Consistent UX** - Plot management feels like other workspace tools
2. **Vim-native navigation** - j/k, Enter, q instead of R functions
3. **Dashboard integration** - All workspace info in one place
4. **Discoverable** - Same patterns users already know from other HUDs
5. **Keyboard-driven** - No mouse needed, efficient workflow

Migration Path

1. Keep existing plot watcher and kitty pane display (working well)
2. Add `:RPlotHUD` as new entry point
3. Refactor `:RPlotGallery` to use HUD styling
4. Add to Dashboard as 6th tab
5. Deprecate old `<LocalLeader>p*` mappings in favor of HUD buffer actions

Open Questions

1. Should kitty pane auto-open when Plot HUD is opened?
2. Should Plot HUD auto-refresh when new plot is created?
3. Should we show thumbnail previews in Vim using ASCII art or sixel?