Summary of BDPG results for single Reserve Selector

Bill Langford

2018-11-25

Contents

This is just a quick and dirty summary of what I think are the main results that I want to put in the first paper. I need this to pare down what I'm doing in the other documents.

This file is cloned from mainResults.Rmd and mainResults_singleRS_RandomSampleExps.Rmd, which were getting too big and unwieldy. In here, I will strip out all references to anything other than a single reserve selector and the most important results. The whole thing should be trivial to rerun for each different reserve selector by just changing the variable `rs_name`.

This file is meant to reduce all of the previous Rmd files that have tons of different forms of results in them down to just the results that are candidates for the paper. Those previous Rmd files include mainResults.Rmd, mainResults_singleRS.Rmd, mainResults_singleRS_RandomSampleExps.Rmd, preliminaryAnalysis.Rmd, and secondCutNotebook.Rmd (and fourthCutNotebook.Rmd?).

0.1 Need to fix:

- Missing plots of rep shortfall vs FRAC_spp_covered

0.2 General knitr setup

Can't remember why this is here, but I'll leave it alone for now in case it was important. Might just have been the default knitr file, however, the commenting out of "message=FALSE" suggests there was a reason for that, which might be related to what's happening in the comments in the "Load necessary files" section below.

1 Choose which reserve selector to analyze

```
#rs_name = "Gurobi"
rs_name = "Marxan_SA"
#rs_name = "Marxan_SA_SS"
#rs_name = "ZL_Backward"
#rs_name = "ZL_Forward"
#rs_name = "SR_Backward"
#rs_name = "SR_Forward"
#rs_name = "UR_Backward"
#rs_name = "UR_Forward"
```

2 Load necessary libraries

```
    #  Note that "message=FALSE" is necessary for this chunk if you want to
    #  generate pdfs.  When the tidyverse package is loaded, it writes puts
    #  out a message that includes some unicode that the normal latex engine
    #  (used to produce the pdf) can't handle and it crashes with the following
    #  message:
    #       ! Package inputenc Error: Unicode character [sqrt symbol goes here] (U+221A)
    #       (inputenc)                not set up for use with LaTeX.
    #  Note that I've also had to remove the sqrt symbol from the error message
    #  when embedding it in the comment here, because even inside the comment,
    #  latex tried to render that and crashed.
    #  More information about this can be found at:
    #    - https://community.rstudio.com/t/tidyverse-1-2-1-knitting-to-pdf-issue/2880/4
    #    - https://community.rstudio.com/t/cant-render-tidyverse-1-2-startup-message-in-latex/2811/5
    #    - https://chrisbeeley.net/?p=1037

library (tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
## Warning: package 'tidyr' was built under R version 3.4.4
## Warning: package 'purrr' was built under R version 3.4.4
## Warning: package 'dplyr' was built under R version 3.4.4
## Warning: package 'stringr' was built under R version 3.4.4
```

3 Load data

3.1 Define functions to read and write input and output data

4 Set file paths

NOTE: Leaving some of the old file paths in here for the moment, in case I need to go back to those results and want to know where they're stored.

```
#================================================================================

# # # base_path = "/Users/bill/D/Projects/ProblemDifficulty/Results/bdpg_20_variants_all_rs_easy_base/b
#
#     #  Easy
# base_path = "/Users/bill/D/Projects/ProblemDifficulty/Results/bdpg_20_variants_all_rs_easy_base_2nd_a
#
#     #  Hard
# base_path = "/Users/bill/D/Projects/ProblemDifficulty/Results/bdpg_20_variants_all_rs_HARD_base_first_

#     #  Hard - 2, 5, 7.5, 10 % error
# base_path = "/Users/bill/D/Projects/ProblemDifficulty/Results/bdpg_20_variants_all_rs_HARD_base_first_

    #  Combined Easy and Hard (all Easy error levels, all Hard except 15%)
#base_path = "/Users/bill/D/Projects/ProblemDifficulty/Results/FullEasyHardV1/cln_easyHard."

#suffix = ".combined_results.csv"

#----------

base_path = "/Users/bill/D/Projects/ProblemDifficulty/Results/ExpGen6basicVariants/Combined_batches_1_2_

suffix = ".csv"
```

4.1 Load combined Easy/Hard input data

```
easyHard_df    = load_input_data (rs_name, base_path, suffix)
```

4.2 Reduce full data set down to working data subset

Reduce the data set to just APP WRAP problems and examples with no input cost error.

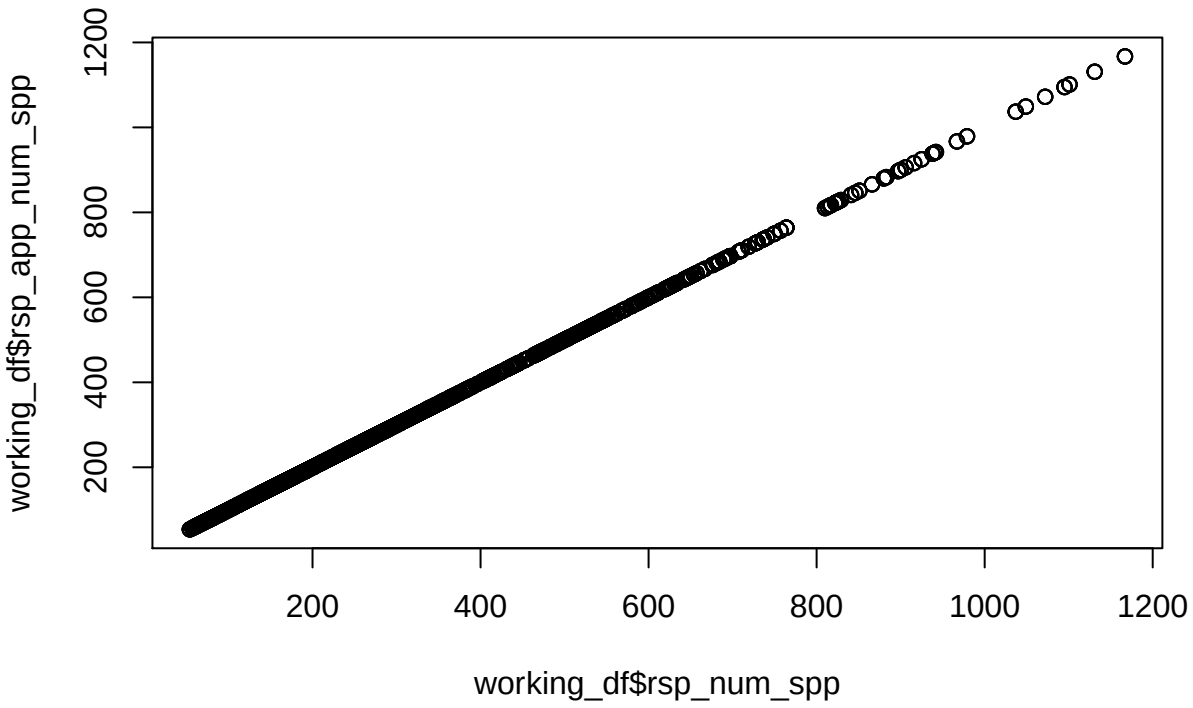
2018 11 25 - NOTE that previous versions of this section didn't include many of the metrics from the bipartite library (since I had forgotten to include their calculation when doing the easy/hard experiments). They also didn't include the various timing results, which I think may not be useful in looking for predicting problem difficulty and output error.

4.2.0.1 Check whether there are problems where COR and APP spp cts are not identical

```
unequals = which (working_df$rsp_num_spp != working_df$rsp_app_num_spp)
cat ("\nNumber of correct and apparent spp pairs that are not the same = ",
     length(unequals), "\n")
```

```
##
## Number of correct and apparent spp pairs that are not the same =  0
```

```
plot (working_df$rsp_num_spp, working_df$rsp_app_num_spp)
```



---

5 Plot results

5.1 Input error vs. output error

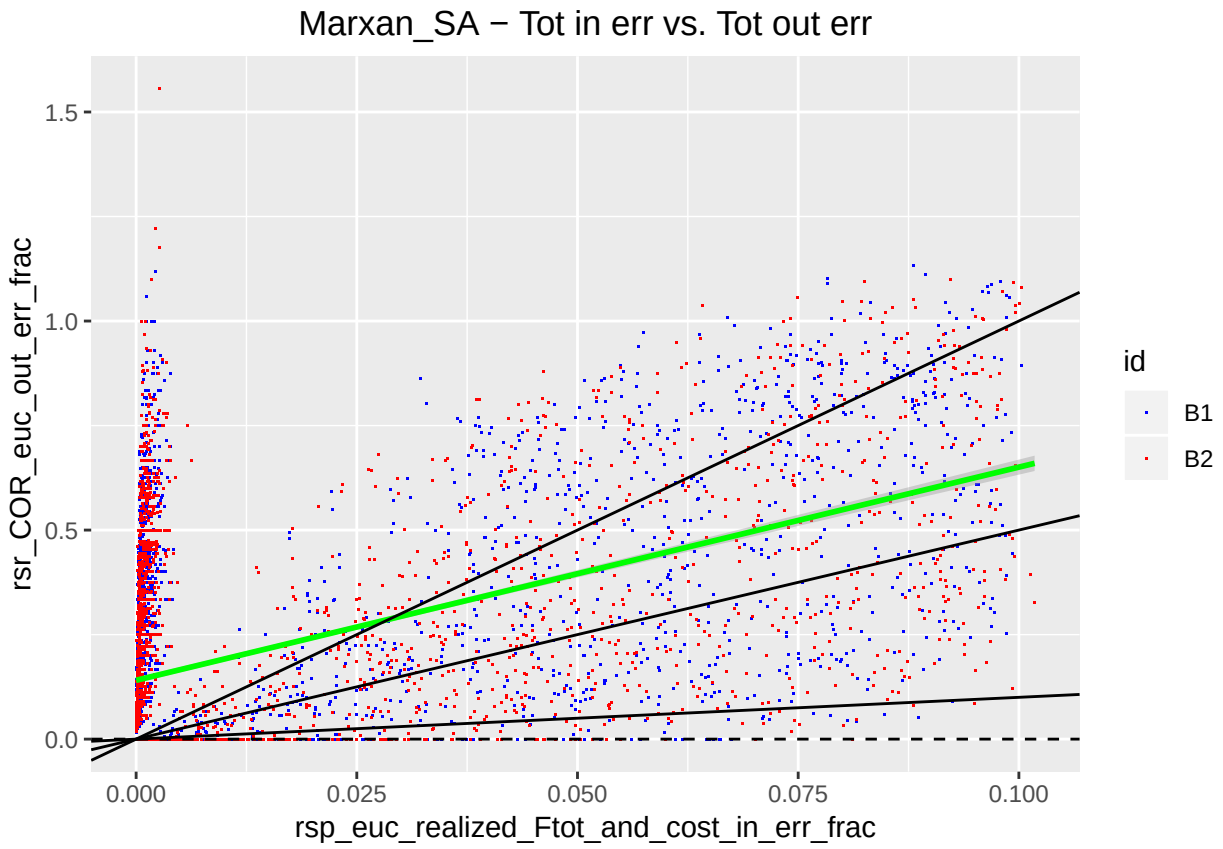
- Generated hard problems were much harder than generated easy problems, as predicted
  - This statement came from when this file was analyzing the easy/hard data sets. Not sure whether it is so easily expressed in this particular document. Will have to look into how to show that here.
- There is lots of variation in problem difficulty at every level of input error
- Solution quality under uncertainty is not good, regardless of optimizer
- Not much difference in reserve selectors
  - Smoothers were slightly better in median (ZL and SA_SS)
- Possible difference for Simple Richness on difficult problem?
- Also need to add plots for all output error fractions values below 1.

5.1.1 Define plotting functions for input error vs. output error

5.1.2 Plot the input error vs output error

5.1.2.1 Full data set

```
gg_eucInTot_vs_eucOutTot_all_on_1 (rs_name, easyHard_df, ref_y = 0
#                                    , shape_type = 1, alpha_level = 1
                                  )
```



5.1.2.2 Full data set, just values that are not huge errors

Will comment this out for the moment, since I'm not sure I want to plot these anymore. They seem to not apply for gurobi and Marxan_SA, since they don't have such huge errors as some of the other reserve selectors. May just need to have some of these outputs be dependent on which reserve selector you're looking at.

6 {r}

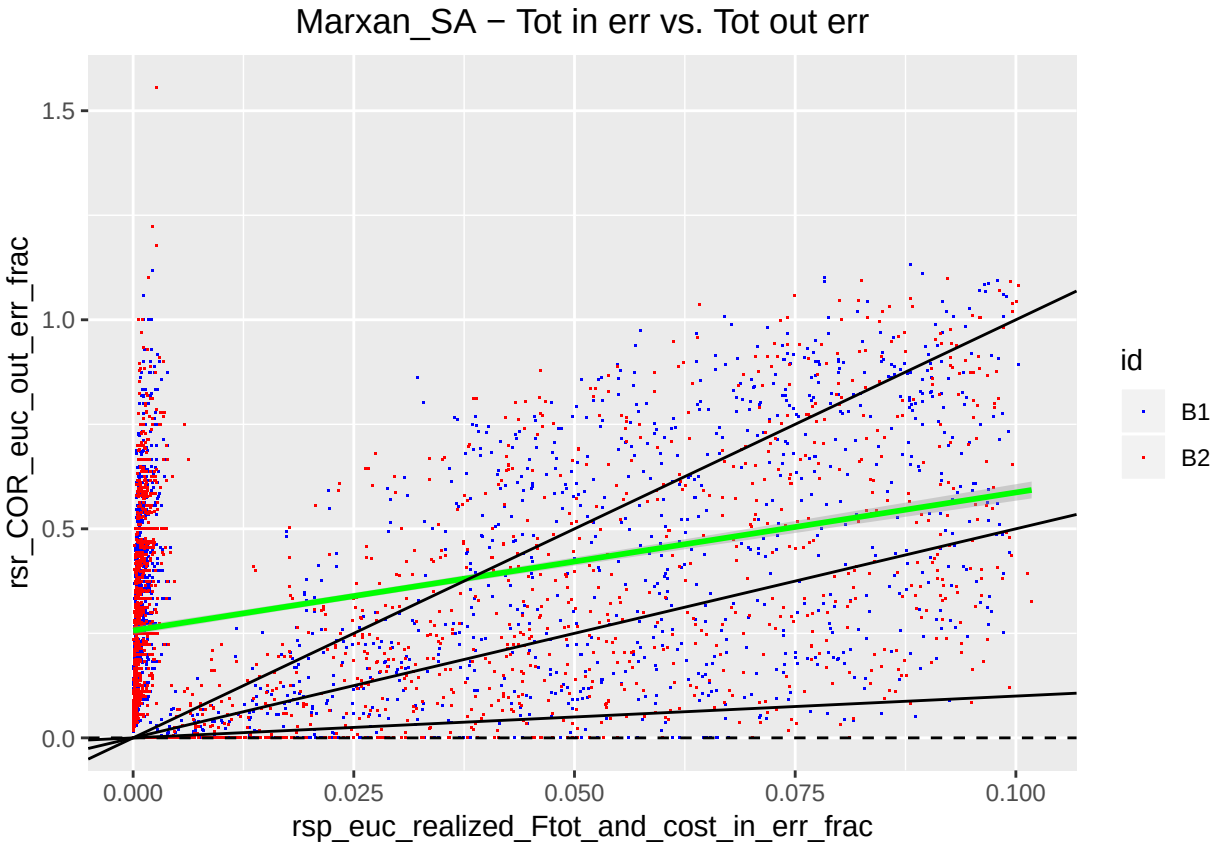
easyHard_df %>% filter (rsr_COR_euc_out_err_frac <= 1) -> easyHard_not_huge_out_err_df gg_eucInTot_vs_eucOutTot_all_on_1 (rs_name, easyHard_not_huge_out_err_df, ref_y = 0 , shape_type = ".", alpha_level = 1)

‘‘‘

6.0.0.1 Working data set

Plot the same results for working data set that only includes APP WRAP problems that have no input cost error. Note that this changes the scaling and the fit lines.

```
gg_eucInTot_vs_eucOutTot_all_on_1 (rs_name, working_df, ref_y = 0
#, shape_type = 1, alpha_level = 1/20
                                  )
```



6.0.0.2 Working data set, just values that are not huge errors

Will comment this out for the moment, since I'm not sure I want to plot these anymore. They seem to not apply for gurobi and Marxan_SA, since they don't have such huge errors as some of the other reserve selectors. May just need to have some of these outputs be dependent on which reserve selector you're looking at.

7 {r}

working_df %>% filter (rsr_COR_euc_out_err_frac <= 1) -> working_not_huge_out_err_df gg_eucInTot_vs_eucOutTot_all_on_1 (rs_name, working_not_huge_out_err_df, ref_y = 0 #, shape_type = ".", alpha_level = 1/20) "'

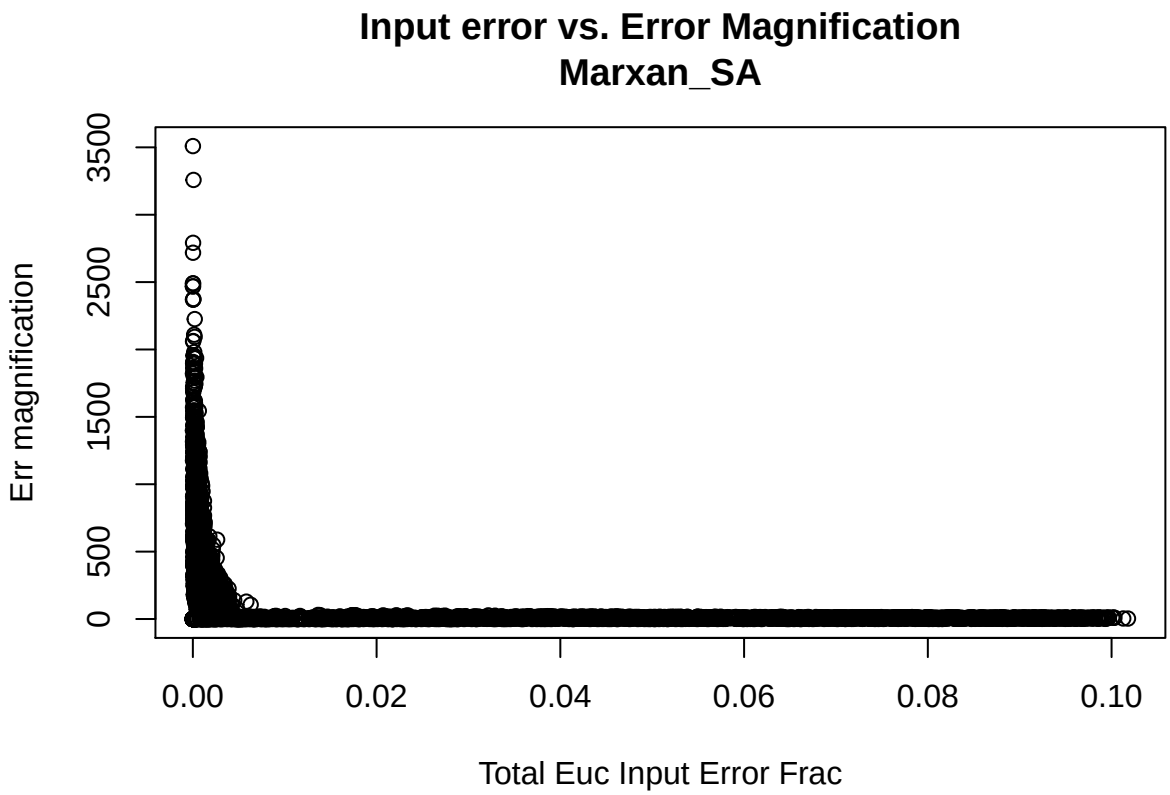
7.1 Error magnification

7.1.1 Optimization often amplifies input error

Not sure this needs a separate plot, since the 10:1, 5:1, and 1:1 lines show it so well and the actual exact numbers aren't that important.

7.1.1.1 Show all magnification values first (to show full range)

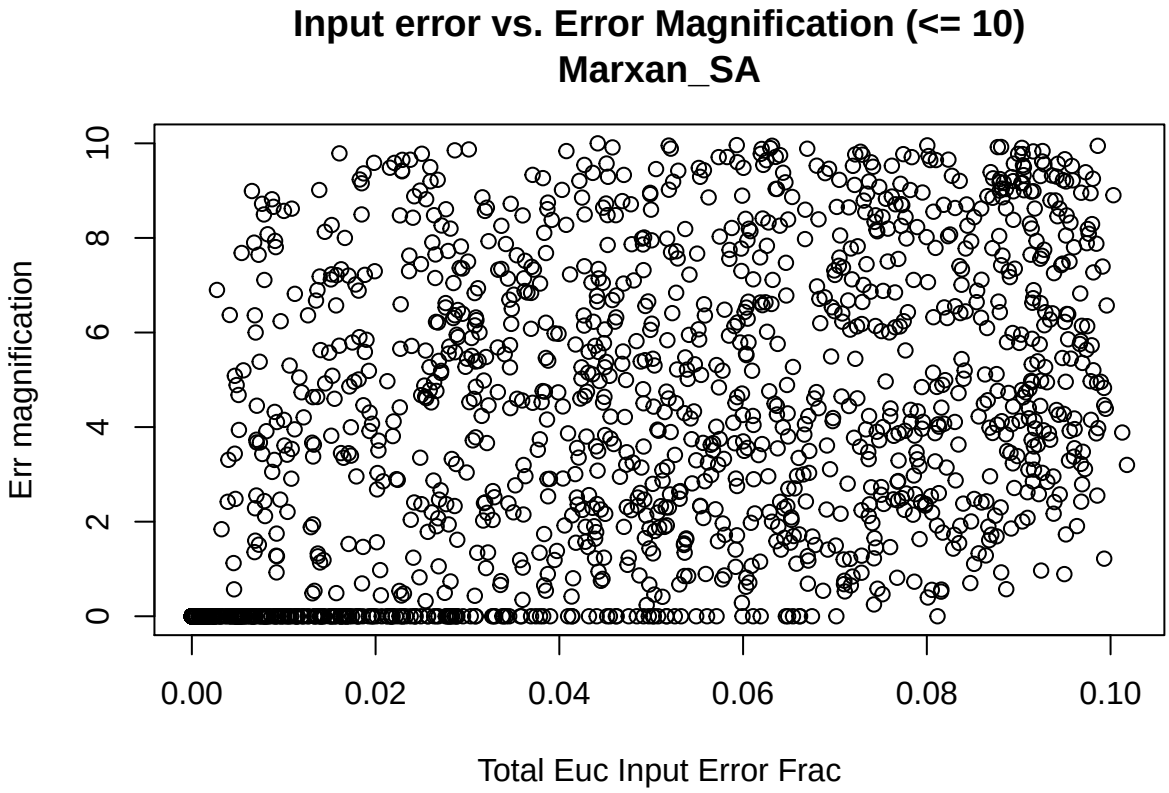
```
plot (working_df$rsp_euc_realized_Ftot_and_cost_in_err_frac,
      working_df$err_mag,
      main=paste0 ("Input error vs. Error Magnification\n", rs_name),
      xlab="Total Euc Input Error Frac",
      ylab="Err magnification")
```



7.1.1.2 Show all magnification values below 10

This gets rid of all the extreme values that result at tiny, tiny input errors.

```
idxs_of_not_huge_err_mags = which (working_df$err_mag <= 10)
plot (working_df$rsp_euc_realized_Ftot_and_cost_in_err_frac [idxs_of_not_huge_err_mags],
      working_df$err_mag [idxs_of_not_huge_err_mags],
      main=paste0 ("Input error vs. Error Magnification (<= 10)\n", rs_name),
      xlab="Total Euc Input Error Frac",
      ylab="Err magnification")
```

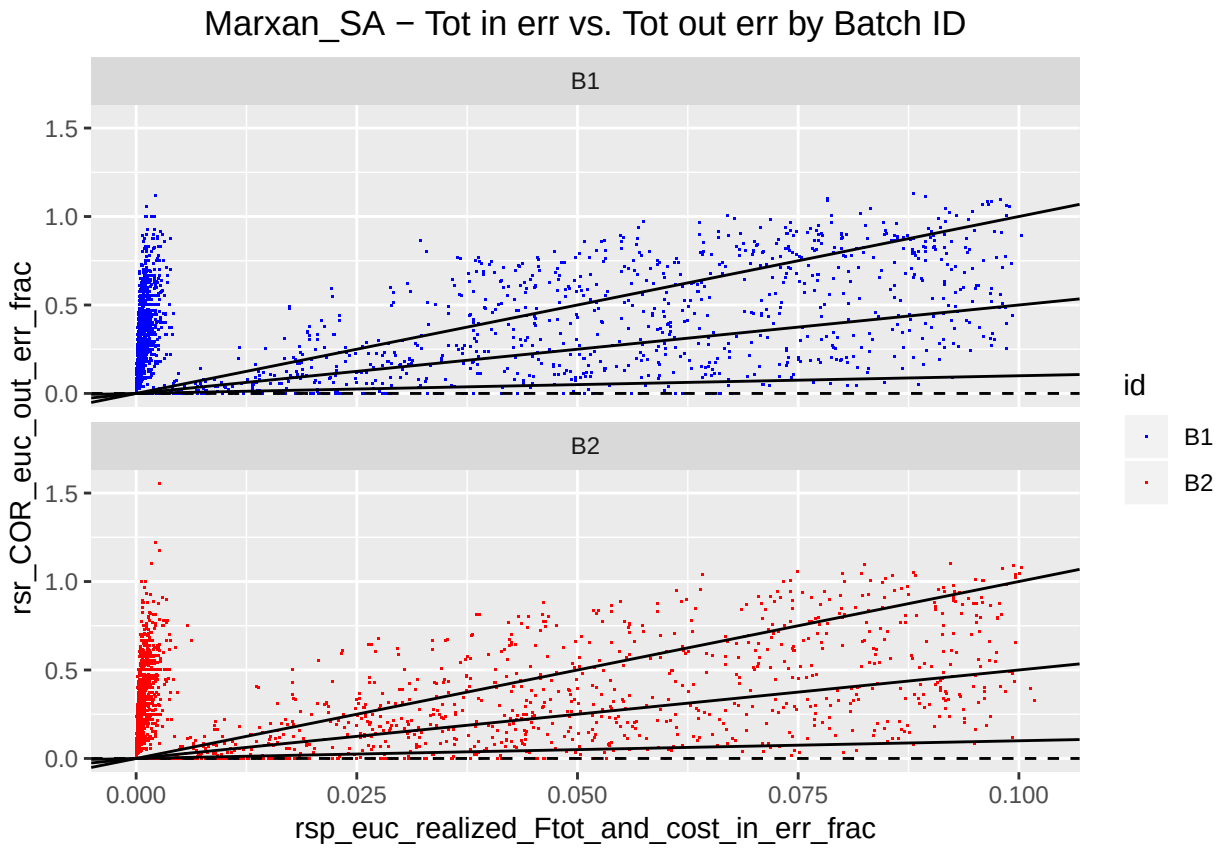


7.2 Different types of input error produced different amounts of output error

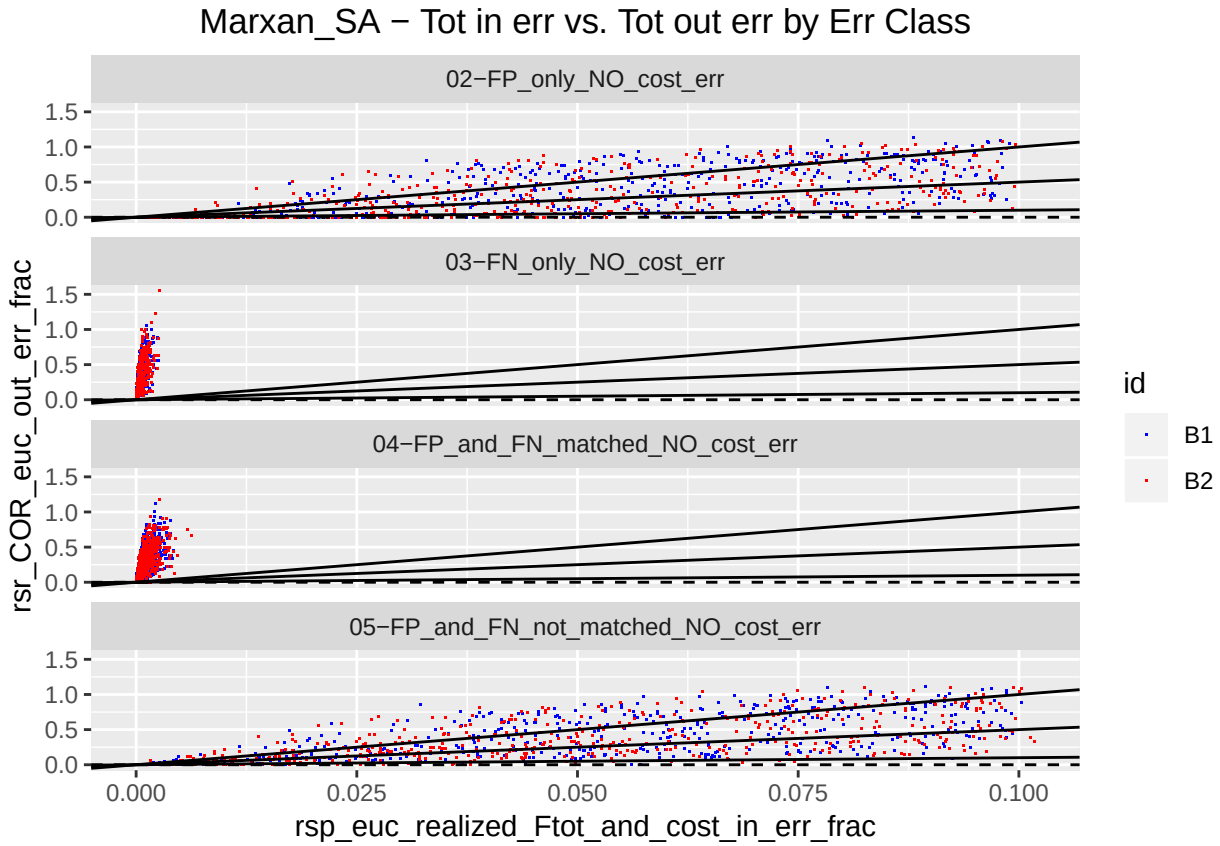
- Uniform random cost error had almost no impact anywhere
- FN produced . . .
- FP produced . . .
- Mixed was determined by FN or FP dominance

NOTE: This plot needs to be redone and broken apart. The facetting done here makes it hard to see what's going on for the FN plots because everything there is compressed into one tiny vertical band. That's because the x-axis is showing total input error rather than just FN error and total input error is small when FN is the dominant form of error. If you just select out the FN-dominant errors, then plot, it will rescale to a much more appropriate x-axis.

```
gg_eucInTot_vs_eucOutTot_facetted_by_id (rs_name, working_df, ref_y = 0)
```




```
#####gg_eucInTot_vs_eucOutTot_facetted_by_err_label (rs_name, easyHard_df, ref_y = 0)
gg_eucInTot_vs_eucOutTot_facetted_by_err_label (rs_name, working_df, ref_y = 0)
```




```
# easyHard_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_eucInTot_vs_eucOutTot_facetted_by_err_label (rs_name, easyHard_not_huge_out_err_df, ref_y = 0)
# working_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_eucInTot_vs_eucOutTot_facetted_by_err_label (rs_name, working_not_huge_out_err_df, ref_y = 0)
```

7.3 FN-dominated input errors

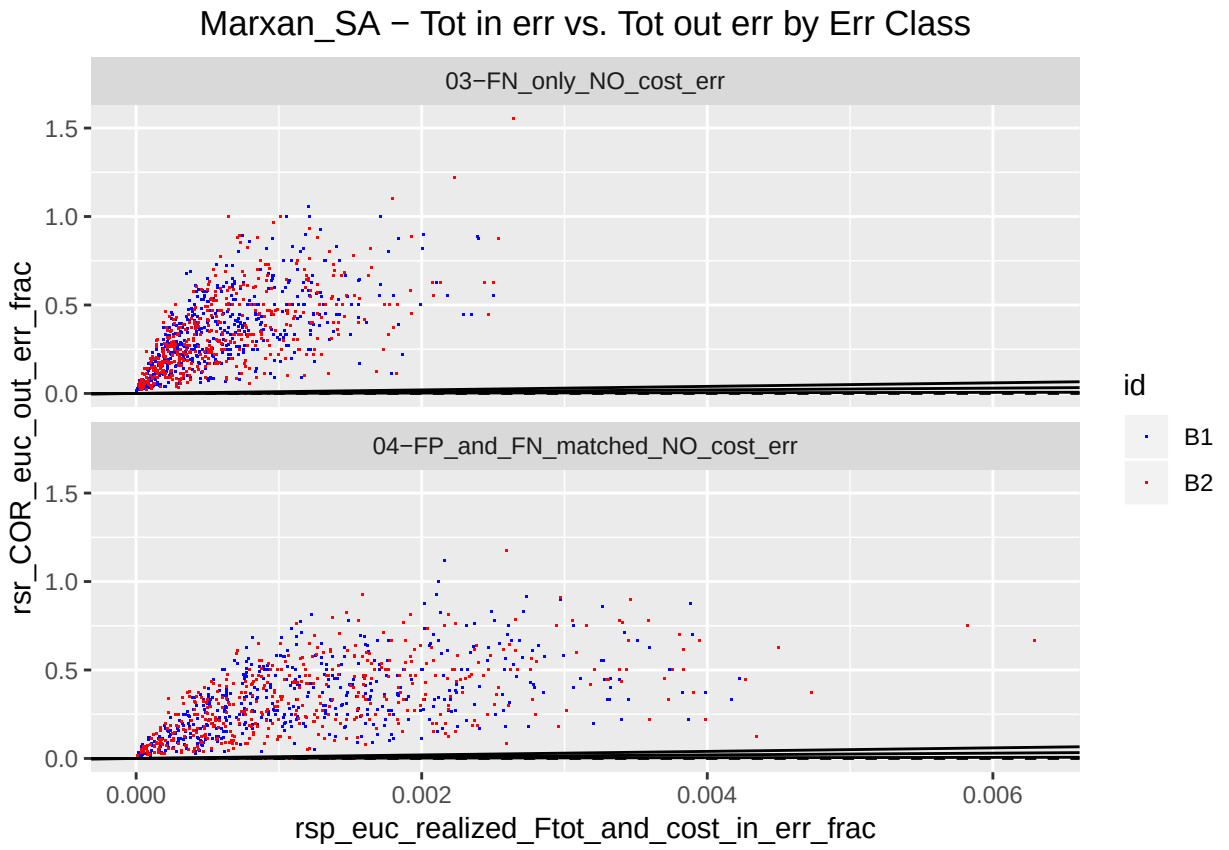
```
#=============================================================================

gg_eucInTot_vs_eucOutTot_facetted_by_FN_dominance <- function (rs_name,
                                                               sorted_msa_tib,
                                                               ref_y = 0
                                                               , shape_type = "."
                                                               , alpha_level = 1
                                                               )
    {
    ggplot (data = sorted_msa_tib) +
      geom_point (mapping = aes(x = rsp_euc_realized_Ftot_and_cost_in_err_frac,
                                y = rsr_COR_euc_out_err_frac,
    #                            color = rsp_combined_err_label)) +
    #                             color = id
    # ),
    # shape="."
```

```
    # ) +
 # color = id
 #    ),
 #    shape="."
 #    ) +
 color = id
    ),
    shape="."
    ) +


    #scale_color_viridis(discrete=TRUE) +
    #scale_color_brewer(palette="Set1") +
    scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

    ggtitle (paste0 (rs_name, " - Tot in err vs. Tot out err by Err Class")) +
    theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

      facet_wrap (~ rsp_combined_err_label, nrow = 2) +
      geom_hline (yintercept = ref_y, linetype="dashed",
                  color = "black", size=0.5) +
      geom_abline (intercept=0, slope=1  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=5  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=10  #, linetype, color, size
                   )
    }

#=============================================================================

gg_rsp_realized_FN_rate_vs_eucOutTot_facetted_by_FN_dominance <- function (rs_name,
                                                                           sorted_msa_tib,
                                                                           ref_y = 0
                                                                           , shape_type = "."
                                                                           , alpha_level = 1
                                                                           )
    {
    ggplot (data = sorted_msa_tib) +
      geom_point (mapping = aes(x = rsp_realized_FN_rate,
                                y = rsr_COR_euc_out_err_frac,
    #                            color = rsp_combined_err_label)) +
    #                             color = id
    # ),
    # shape="."
    # ) +
 # color = id
 #    ),
 #    shape="."
 #    ) +
 color = id
    ),
    shape="."
```

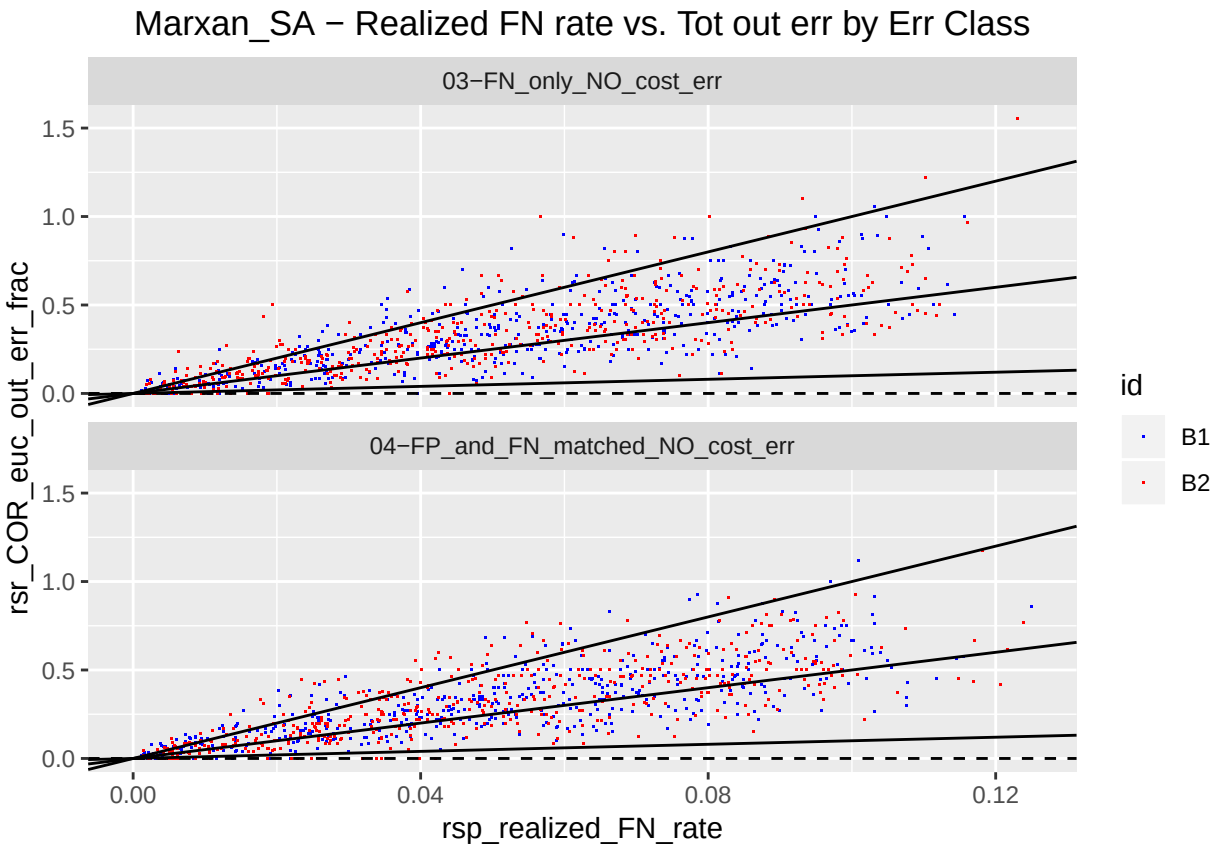
```
    ) +


    #scale_color_viridis(discrete=TRUE) +
    #scale_color_brewer(palette="Set1") +
    scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

    ggtitle (paste0 (rs_name, " - Realized FN rate vs. Tot out err by Err Class")) +
    theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

      facet_wrap (~ rsp_combined_err_label, nrow = 2) +
      geom_hline (yintercept = ref_y, linetype="dashed",
                  color = "black", size=0.5) +
      geom_abline (intercept=0, slope=1  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=5  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=10  #, linetype, color, size
                   )
    }
```

```
working_df %>%
    filter (rsp_combined_err_label == "03-FN_only_NO_cost_err" |
            rsp_combined_err_label == "04-FP_and_FN_matched_NO_cost_err"
) -> FN_dominant_working_df
```

```
gg_eucInTot_vs_eucOutTot_facetted_by_FN_dominance (rs_name,
                                                   FN_dominant_working_df,
                                                   ref_y = 0)
```




```
gg_rsp_realized_FN_rate_vs_eucOutTot_facetted_by_FN_dominance (rs_name,
                                                               FN_dominant_working_df,
                                                               ref_y = 0)
```

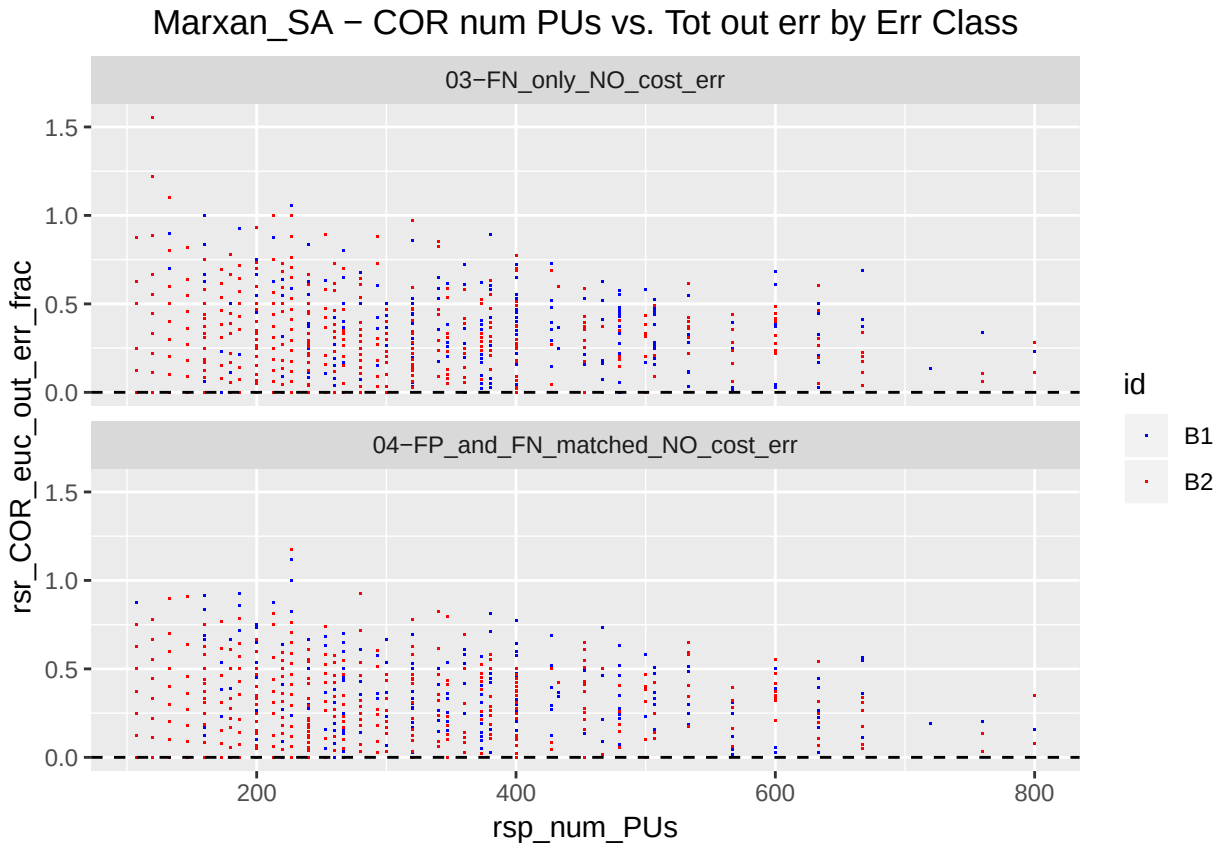


7.3.1 Do things like num_spp predict total output error under FN-dominated error?

```
#==============================================================================

gg_rsp_num_PUs_vs_eucOutTot_facetted_by_FN_dominance <- function (rs_name,
                                                                  sorted_msa_tib,
                                                                  ref_y = 0
                                                                  , shape_type = "."
                                                                  , alpha_level = 1
                                                                  )
    {
    ggplot (data = sorted_msa_tib) +
      geom_point (mapping = aes(x = rsp_num_PUs,
                                y = rsr_COR_euc_out_err_frac,
    #                            color = rsp_combined_err_label)) +
    #                             color = id
    # ),
    # shape="."
    # ) +
 # color = id
 #    ),
 #    shape="."
 #    ) +
 color = id
    ),
```

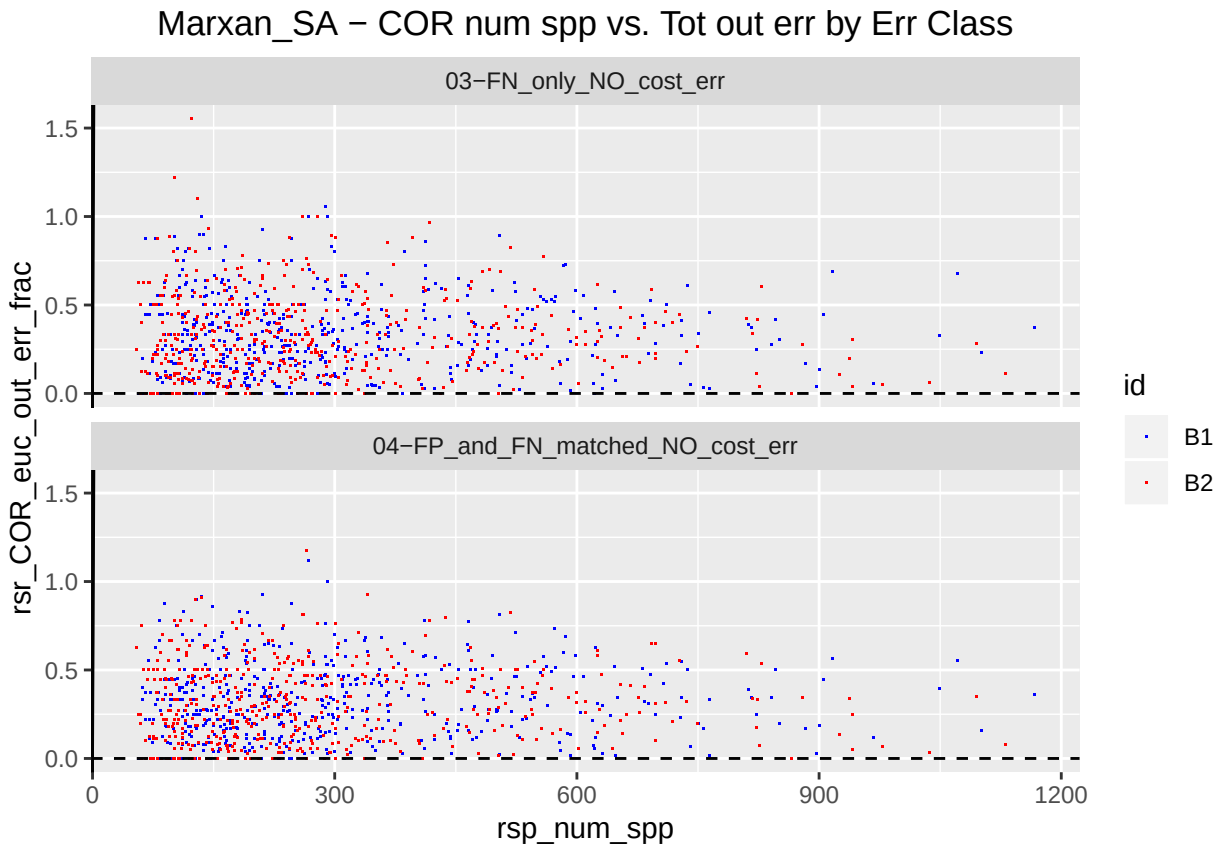
```
    shape="."
    ) +


    #scale_color_viridis(discrete=TRUE) +
    #scale_color_brewer(palette="Set1") +
    scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

    ggtitle (paste0 (rs_name, " - COR num PUs vs. Tot out err by Err Class")) +
    theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

      facet_wrap (~ rsp_combined_err_label, nrow = 2) +
      geom_hline (yintercept = ref_y, linetype="dashed",
                  color = "black", size=0.5) +
      geom_abline (intercept=0, slope=1  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=5  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=10  #, linetype, color, size
                   )
    }
```

```
gg_rsp_num_PUs_vs_eucOutTot_facetted_by_FN_dominance (rs_name,
                                                      FN_dominant_working_df,
                                                      ref_y = 0)
```



```
#==========================================================================

gg_rsp_num_spp_vs_eucOutTot_facetted_by_FN_dominance <- function (rs_name,
                                                                  sorted_msa_tib,
                                                                  ref_y = 0
                                                                  , shape_type = "."
                                                                  , alpha_level = 1
                                                                  )
    {
    ggplot (data = sorted_msa_tib) +
      geom_point (mapping = aes(x = rsp_num_spp,
                                y = rsr_COR_euc_out_err_frac,
    #                            color = rsp_combined_err_label)) +
    #                             color = id
    # ),
    # shape="."
    # ) +
 # color = id
 #    ),
 #    shape="."
 #    ) +
 color = id
    ),
    shape="."
    ) +


    #scale_color_viridis(discrete=TRUE) +
    #scale_color_brewer(palette="Set1") +
    scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

    ggtitle (paste0 (rs_name, " - COR num spp vs. Tot out err by Err Class")) +
    theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

      facet_wrap (~ rsp_combined_err_label, nrow = 2) +
      geom_hline (yintercept = ref_y, linetype="dashed",
                    color = "black", size=0.5) +
      geom_abline (intercept=0, slope=1  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=5  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=10  #, linetype, color, size
                   )
    }
```

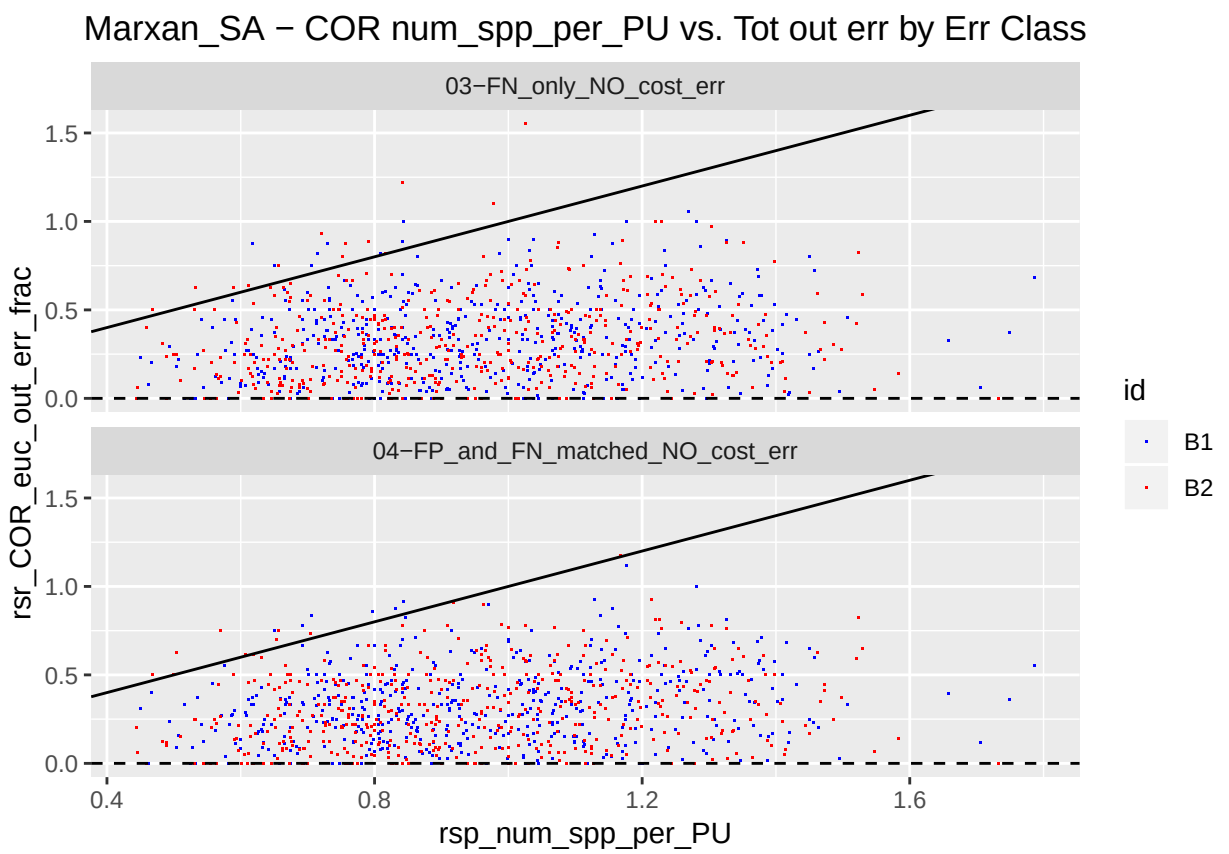
```
gg_rsp_num_spp_vs_eucOutTot_facetted_by_FN_dominance (rs_name,
                                                      FN_dominant_working_df,
                                                      ref_y = 0)
```




```
#================================================================================

gg_rsp_num_spp_per_PU_vs_eucOutTot_facetted_by_FN_dominance <- function (rs_name,
                                                                         sorted_msa_tib,
                                                                         ref_y = 0
                                                                         , shape_type = "."
                                                                         , alpha_level = 1
                                                                         )
    {
    ggplot (data = sorted_msa_tib) +
      geom_point (mapping = aes(x = rsp_num_spp_per_PU,
                                y = rsr_COR_euc_out_err_frac,
    #                            color = rsp_combined_err_label)) +
    #                             color = id
    # ),
    # shape="."
    # ) +
 # color = id
 #    ),
 #    shape="."
 #    ) +
 color = id
    ),
    shape="."
    ) +
```

```
    #scale_color_viridis(discrete=TRUE) +
    #scale_color_brewer(palette="Set1") +
    scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

    ggtitle (paste0 (rs_name, " - COR num_spp_per_PU vs. Tot out err by Err Class")) +
    theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

      facet_wrap (~ rsp_combined_err_label, nrow = 2) +
      geom_hline (yintercept = ref_y, linetype="dashed",
                  color = "black", size=0.5) +
      geom_abline (intercept=0, slope=1  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=5  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=10  #, linetype, color, size
                   )
    }
```

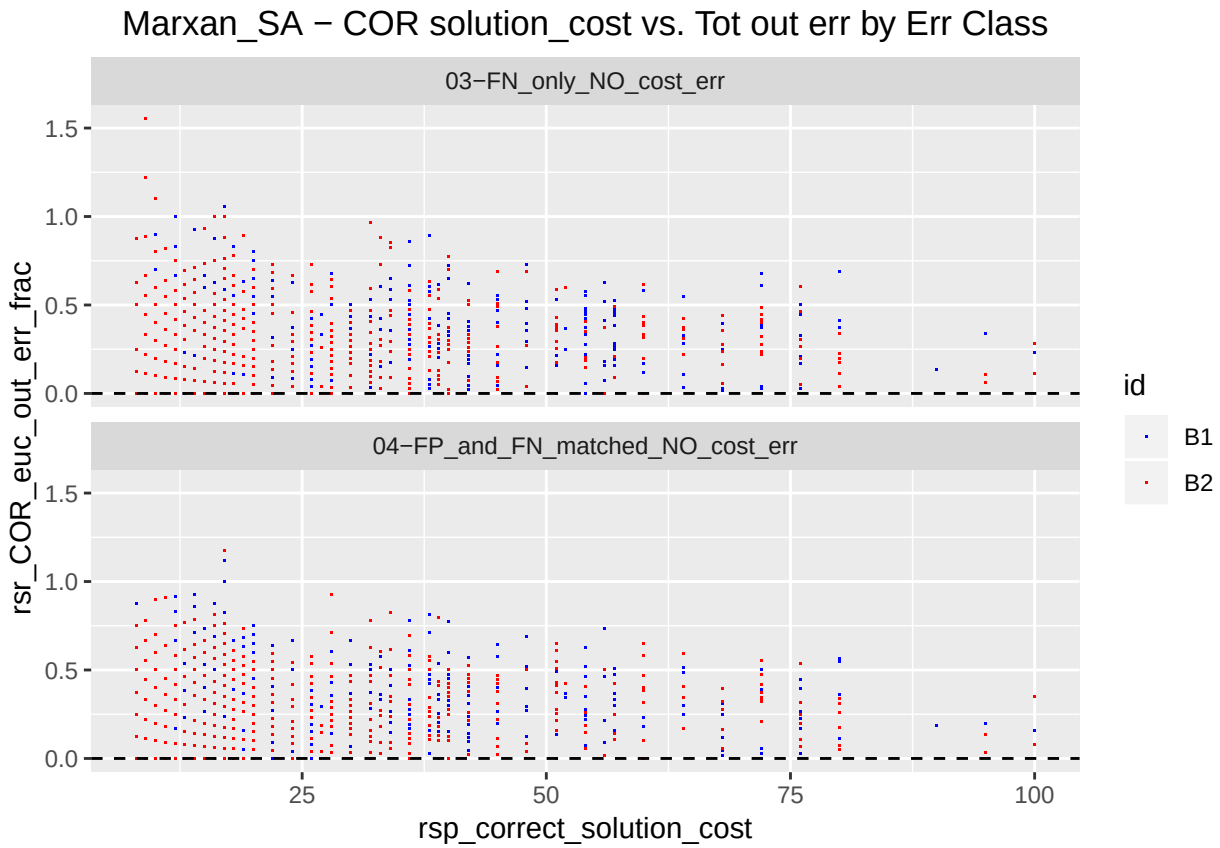
```
gg_rsp_num_spp_per_PU_vs_eucOutTot_facetted_by_FN_dominance (rs_name,
                                                             FN_dominant_working_df,
                                                             ref_y = 0)
```




```
#=============================================================================

gg_rsp_correct_solution_cost_vs_eucOutTot_facetted_by_FN_dominance <- function (rs_name,
                                                                  sorted_msa_tib,
                                                                  ref_y = 0
```

```
                                                          , shape_type = "."
                                                          , alpha_level = 1
                                                          )
   {
   ggplot (data = sorted_msa_tib) +
     geom_point (mapping = aes(x = rsp_correct_solution_cost,
                               y = rsr_COR_euc_out_err_frac,
   #                            color = rsp_combined_err_label)) +
   #                             color = id
   # ),
   # shape="."
   # ) +
# color = id
#    ),
#    shape="."
#    ) +
color = id
   ),
   shape="."
   ) +


   #scale_color_viridis(discrete=TRUE) +
   #scale_color_brewer(palette="Set1") +
   scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

   ggtitle (paste0 (rs_name, " - COR solution_cost vs. Tot out err by Err Class")) +
   theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

     facet_wrap (~ rsp_combined_err_label, nrow = 2) +
     geom_hline (yintercept = ref_y, linetype="dashed",
                 color = "black", size=0.5) +
     geom_abline (intercept=0, slope=1  #, linetype, color, size
                  ) +
     geom_abline (intercept=0, slope=5  #, linetype, color, size
                  ) +
     geom_abline (intercept=0, slope=10  #, linetype, color, size
                  )
   }
```

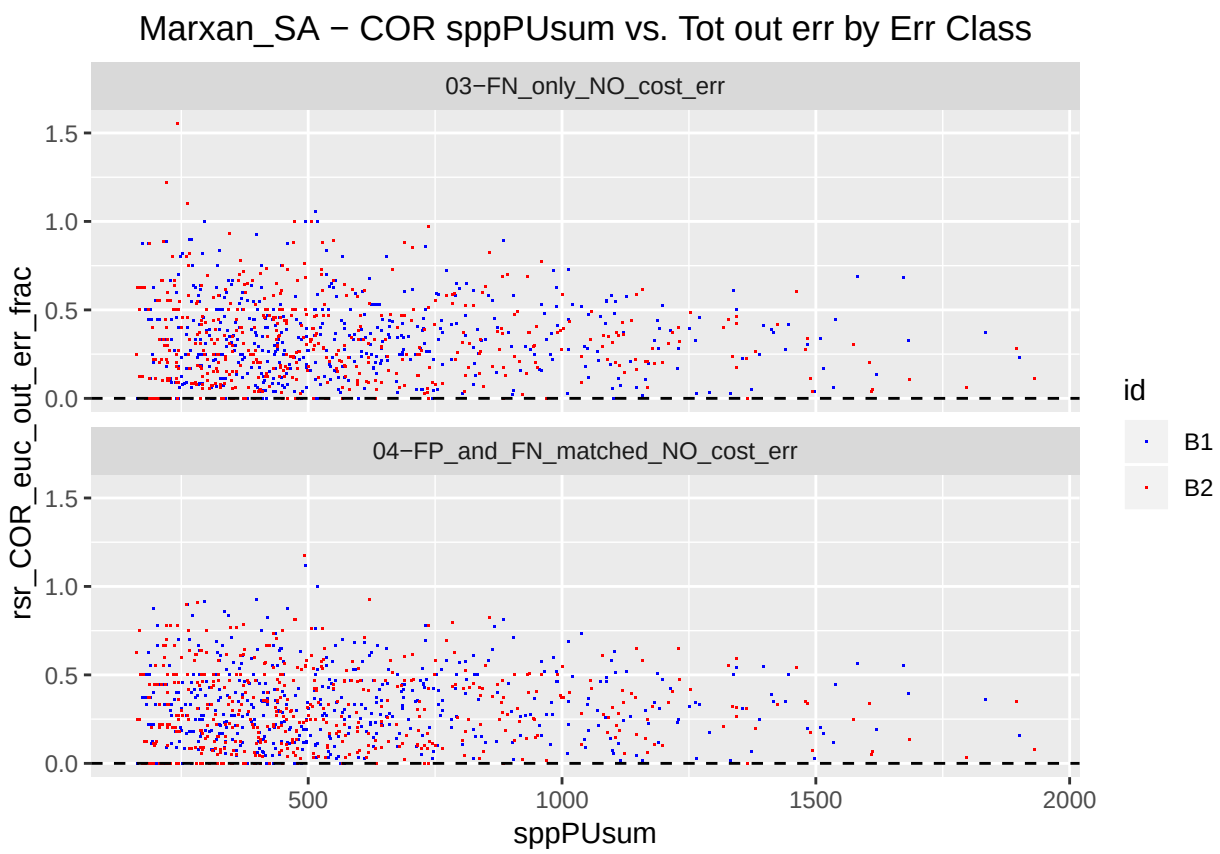
```
gg_rsp_correct_solution_cost_vs_eucOutTot_facetted_by_FN_dominance (rs_name,
                                                    FN_dominant_working_df,
                                                    ref_y = 0)
```




```
#==============================================================================

gg_sppPUsum_vs_eucOutTot_facetted_by_FN_dominance <- function (rs_name,
                                                               sorted_msa_tib,
                                                               ref_y = 0
                                                               , shape_type = "."
                                                               , alpha_level = 1
                                                               )
    {
    ggplot (data = sorted_msa_tib) +
      geom_point (mapping = aes(x = sppPUsum,
                                y = rsr_COR_euc_out_err_frac,
    #                            color = rsp_combined_err_label)) +
    #                             color = id
    # ),
    # shape="."
    # ) +
 # color = id
 #    ),
 #    shape="."
 #    ) +
 color = id
    ),
    shape="."
    ) +
```

```
    #scale_color_viridis(discrete=TRUE) +
    #scale_color_brewer(palette="Set1") +
    scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

    ggtitle (paste0 (rs_name, " - COR sppPUsum vs. Tot out err by Err Class")) +
    theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

      facet_wrap (~ rsp_combined_err_label, nrow = 2) +
      geom_hline (yintercept = ref_y, linetype="dashed",
                  color = "black", size=0.5) +
      geom_abline (intercept=0, slope=1  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=5  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=10  #, linetype, color, size
                   )
    }
```

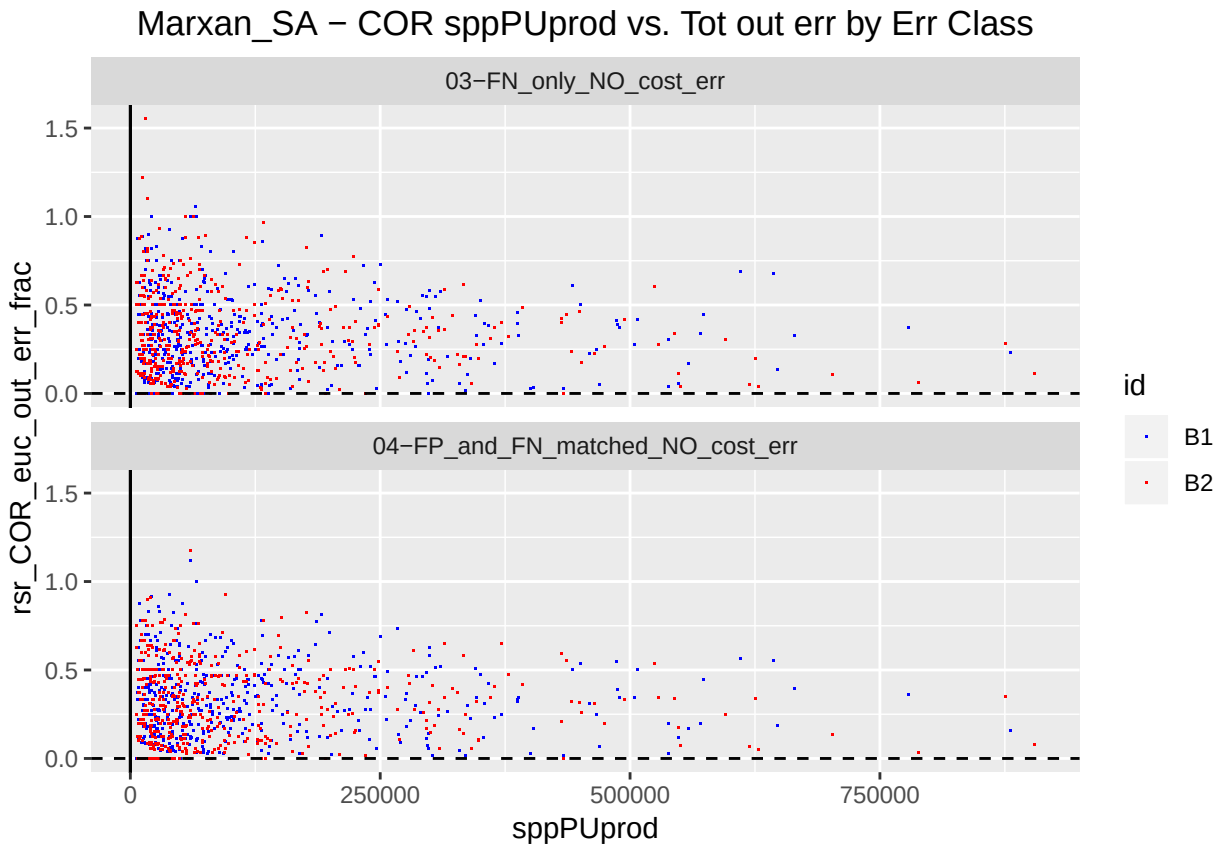
```
gg_sppPUsum_vs_eucOutTot_facetted_by_FN_dominance (rs_name,
                                                   FN_dominant_working_df,
                                                   ref_y = 0)
```




```
#=============================================================================

gg_sppPUprod_vs_eucOutTot_facetted_by_FN_dominance <- function (rs_name,
                                                                sorted_msa_tib,
                                                                ref_y = 0
```

```
                                                          , shape_type = "."
                                                          , alpha_level = 1
                                                          )
   {
   ggplot (data = sorted_msa_tib) +
     geom_point (mapping = aes(x = sppPUprod,
                               y = rsr_COR_euc_out_err_frac,
   #                            color = rsp_combined_err_label)) +
   #                             color = id
   # ),
   # shape="."
   # ) +
# color = id
#    ),
#    shape="."
#    ) +
color = id
   ),
   shape="."
   ) +


   #scale_color_viridis(discrete=TRUE) +
   #scale_color_brewer(palette="Set1") +
   scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

   ggtitle (paste0 (rs_name, " - COR sppPUprod vs. Tot out err by Err Class")) +
   theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

     facet_wrap (~ rsp_combined_err_label, nrow = 2) +
     geom_hline (yintercept = ref_y, linetype="dashed",
                 color = "black", size=0.5) +
     geom_abline (intercept=0, slope=1  #, linetype, color, size
                  ) +
     geom_abline (intercept=0, slope=5  #, linetype, color, size
                  ) +
     geom_abline (intercept=0, slope=10  #, linetype, color, size
                  )
   }
```

```
gg_sppPUprod_vs_eucOutTot_facetted_by_FN_dominance (rs_name,
                                                    FN_dominant_working_df,
                                                    ref_y = 0)
```



7.4 FP-dominated input errors

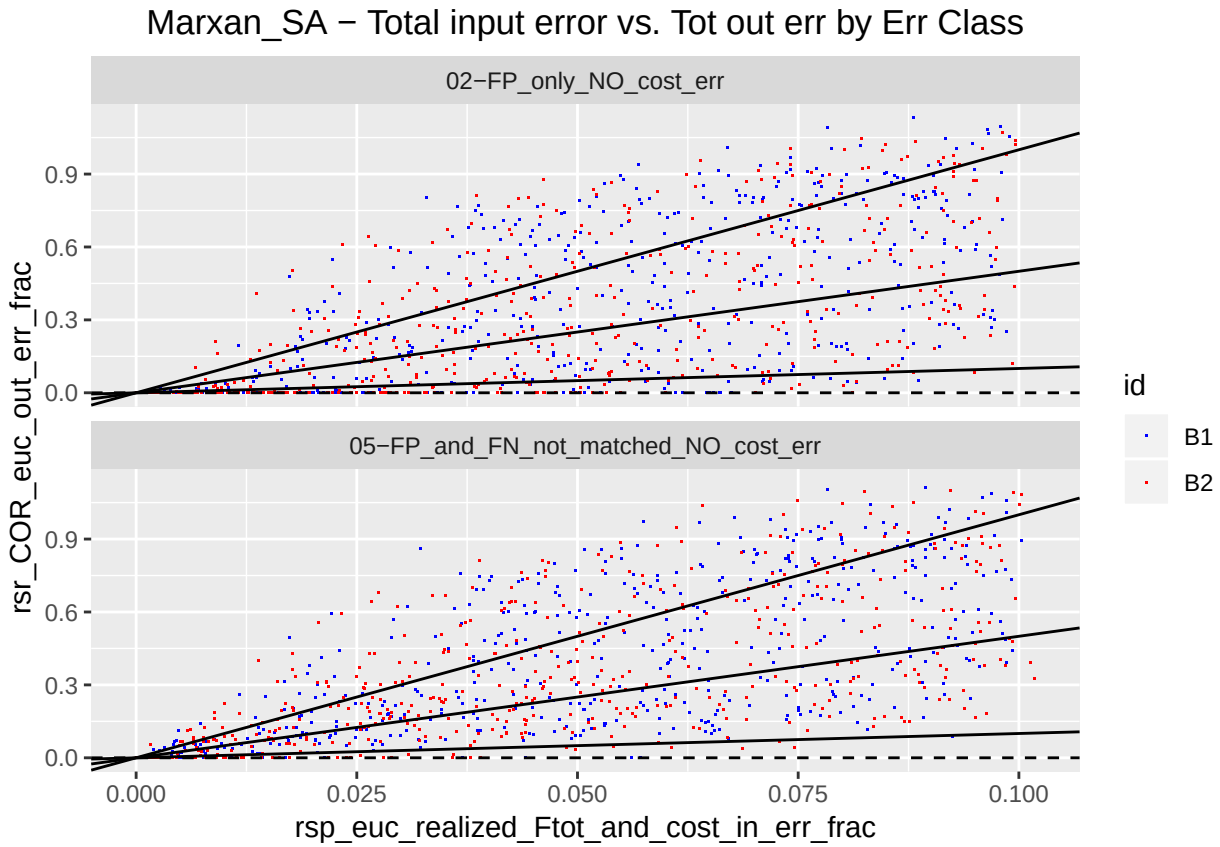
```
working_df %>%
    filter (rsp_combined_err_label == "02-FP_only_NO_cost_err" |
            rsp_combined_err_label == "05-FP_and_FN_not_matched_NO_cost_err"
) -> FP_dominant_working_df
```

```
#==============================================================================

gg_eucInTot_vs_eucOutTot_facetted_by_FP_dominance <- function (rs_name,
                                                               sorted_msa_tib,
                                                               ref_y = 0
                                                               , shape_type = "."
                                                               , alpha_level = 1
                                                               )
    {
    ggplot (data = sorted_msa_tib) +
      geom_point (mapping = aes(x = rsp_euc_realized_Ftot_and_cost_in_err_frac,
                                y = rsr_COR_euc_out_err_frac,
    #                            color = rsp_combined_err_label)) +
    #                             color = id
    # ),
    # shape="."
    # ) +
 # color = id
```

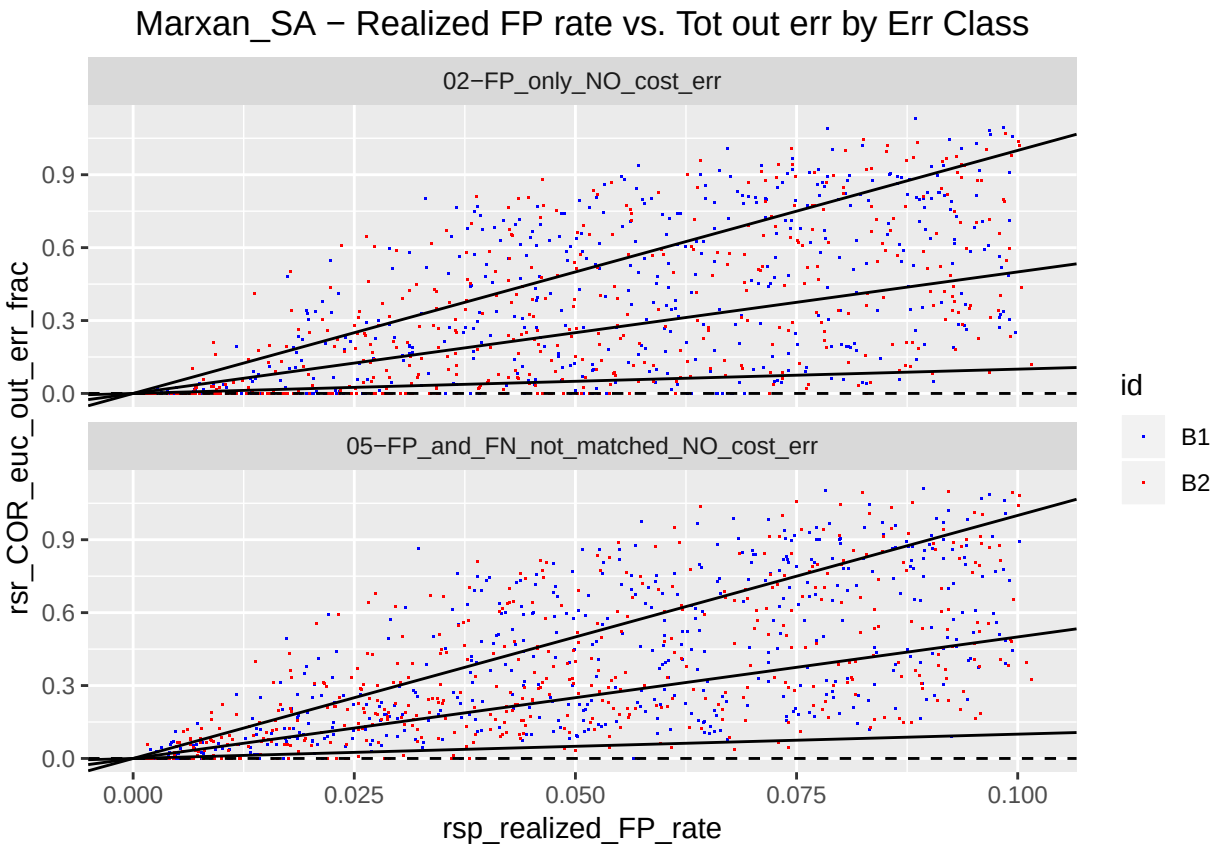
```
#    ),
#    shape="."
#    ) +
color = id
   ),
   shape="."
   ) +


   #scale_color_viridis(discrete=TRUE) +
   #scale_color_brewer(palette="Set1") +
   scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

   ggtitle (paste0 (rs_name, " - Total input error vs. Tot out err by Err Class")) +
   theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

     facet_wrap (~ rsp_combined_err_label, nrow = 2) +
     geom_hline (yintercept = ref_y, linetype="dashed",
                   color = "black", size=0.5) +
     geom_abline (intercept=0, slope=1  #, linetype, color, size
                  ) +
     geom_abline (intercept=0, slope=5  #, linetype, color, size
                  ) +
     geom_abline (intercept=0, slope=10  #, linetype, color, size
                  )
   }

#==============================================================================

gg_rsp_realized_FP_rate_vs_eucOutTot_facetted_by_FP_dominance <- function (rs_name,
                                                                           sorted_msa_tib,
                                                                           ref_y = 0
                                                                           , shape_type = "."
                                                                           , alpha_level = 1
                                                                           )
   {
   ggplot (data = sorted_msa_tib) +
     geom_point (mapping = aes(x = rsp_realized_FP_rate,
                               y = rsr_COR_euc_out_err_frac,
   #                            color = rsp_combined_err_label)) +
   #                             color = id
   # ),
   # shape="."
   # ) +
# color = id
#    ),
#    shape="."
#    ) +
color = id
   ),
   shape="."
   ) +
```

```
    #scale_color_viridis(discrete=TRUE) +
    #scale_color_brewer(palette="Set1") +
    scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

    ggtitle (paste0 (rs_name, " - Realized FP rate vs. Tot out err by Err Class")) +
    theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

      facet_wrap (~ rsp_combined_err_label, nrow = 2) +
      geom_hline (yintercept = ref_y, linetype="dashed",
                  color = "black", size=0.5) +
      geom_abline (intercept=0, slope=1  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=5  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=10  #, linetype, color, size
                   )
    }
```

```
gg_eucInTot_vs_eucOutTot_facetted_by_FP_dominance (rs_name,
                                                   FP_dominant_working_df,
                                                   ref_y = 0)
```




```
gg_rsp_realized_FP_rate_vs_eucOutTot_facetted_by_FP_dominance (rs_name,
                                                   FP_dominant_working_df,
                                                   ref_y = 0)
```

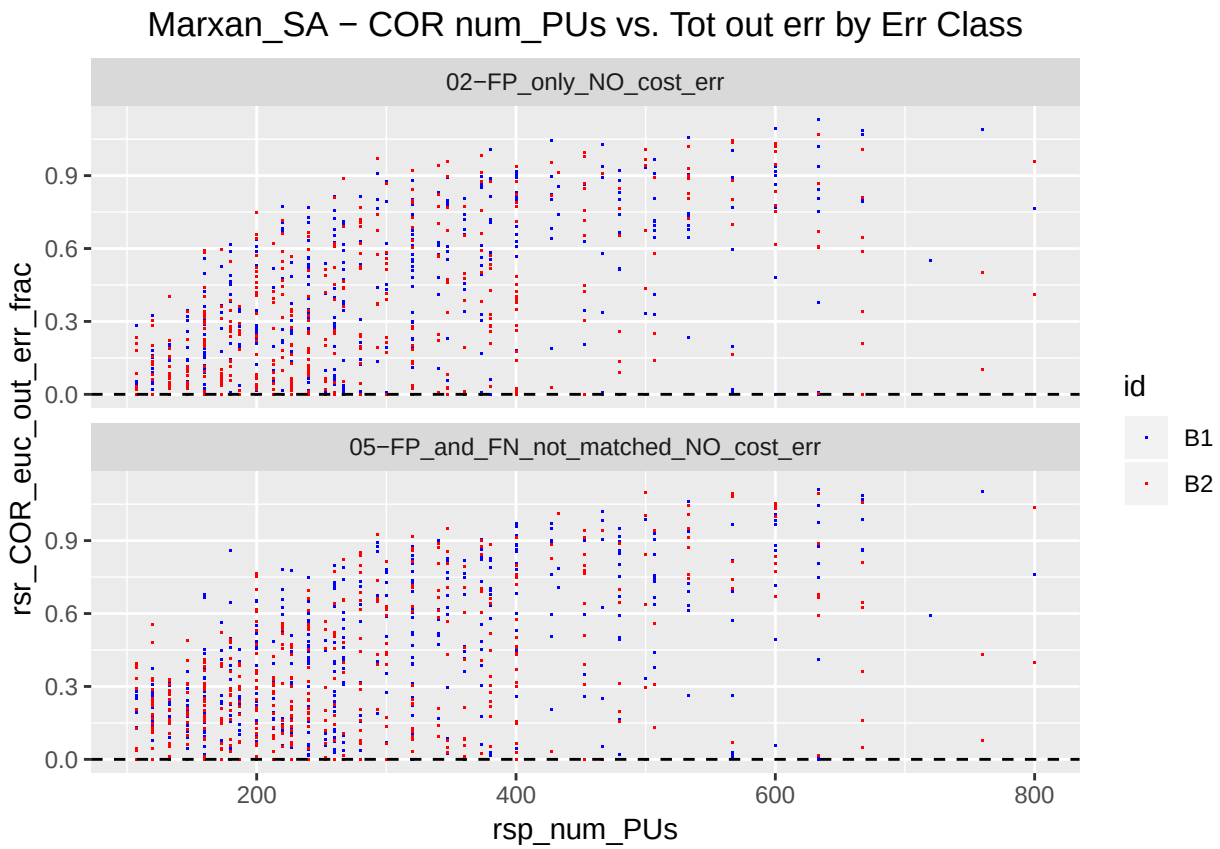


7.4.1 Do things like num_spp predict total output error under FN-dominated error?

```
#==============================================================================

gg_rsp_num_PUs_vs_eucOutTot_facetted_by_FP_dominance <- function (rs_name,
                                                                  sorted_msa_tib,
                                                                  ref_y = 0
                                                                  , shape_type = "."
                                                                  , alpha_level = 1
                                                                  )
    {
    ggplot (data = sorted_msa_tib) +
      geom_point (mapping = aes(x = rsp_num_PUs,
                                y = rsr_COR_euc_out_err_frac,
    #                            color = rsp_combined_err_label)) +
    #                             color = id
    # ),
    # shape="."
    # ) +
 # color = id
 #    ),
 #    shape="."
 #    ) +
 color = id
    ),
```

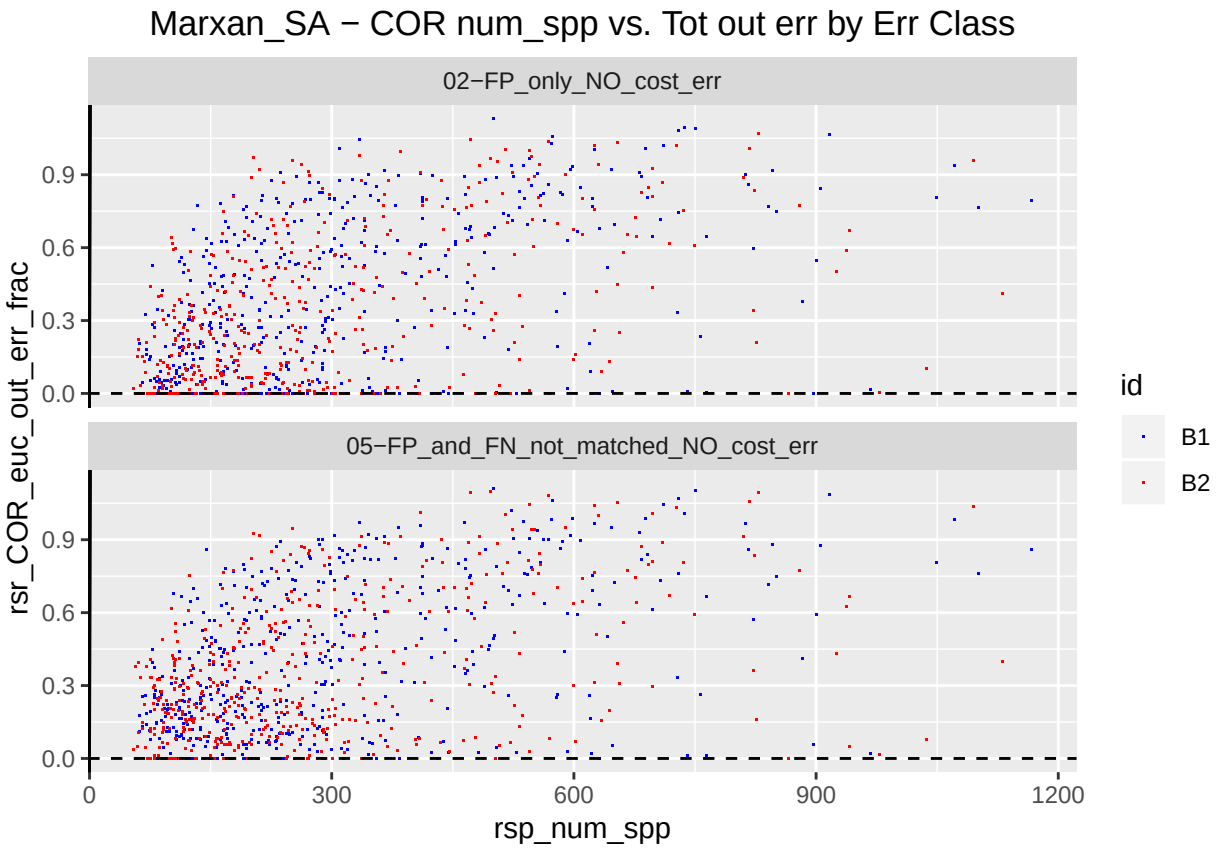
```
    shape="."
    ) +


    #scale_color_viridis(discrete=TRUE) +
    #scale_color_brewer(palette="Set1") +
    scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

    ggtitle (paste0 (rs_name, " - COR num_PUs vs. Tot out err by Err Class")) +
    theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

      facet_wrap (~ rsp_combined_err_label, nrow = 2) +
      geom_hline (yintercept = ref_y, linetype="dashed",
                  color = "black", size=0.5) +
      geom_abline (intercept=0, slope=1  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=5  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=10  #, linetype, color, size
                   )
    }
```

```
gg_rsp_num_PUs_vs_eucOutTot_facetted_by_FP_dominance (rs_name,
                                                      FP_dominant_working_df,
                                                      ref_y = 0)
```



```
#=============================================================================

gg_rsp_num_spp_vs_eucOutTot_facetted_by_FP_dominance <- function (rs_name,
                                                                  sorted_msa_tib,
                                                                  ref_y = 0
                                                                  , shape_type = "."
                                                                  , alpha_level = 1
                                                                  )
    {
    ggplot (data = sorted_msa_tib) +
      geom_point (mapping = aes(x = rsp_num_spp,
                                y = rsr_COR_euc_out_err_frac,
    #                            color = rsp_combined_err_label)) +
    #                             color = id
    # ),
    # shape="."
    # ) +
 # color = id
 #    ),
 #    shape="."
 #    ) +
 color = id
    ),
    shape="."
    ) +


    #scale_color_viridis(discrete=TRUE) +
    #scale_color_brewer(palette="Set1") +
    scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

    ggtitle (paste0 (rs_name, " - COR num_spp vs. Tot out err by Err Class")) +
    theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

      facet_wrap (~ rsp_combined_err_label, nrow = 2) +
      geom_hline (yintercept = ref_y, linetype="dashed",
                    color = "black", size=0.5) +
      geom_abline (intercept=0, slope=1  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=5  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=10  #, linetype, color, size
                   )
    }
```

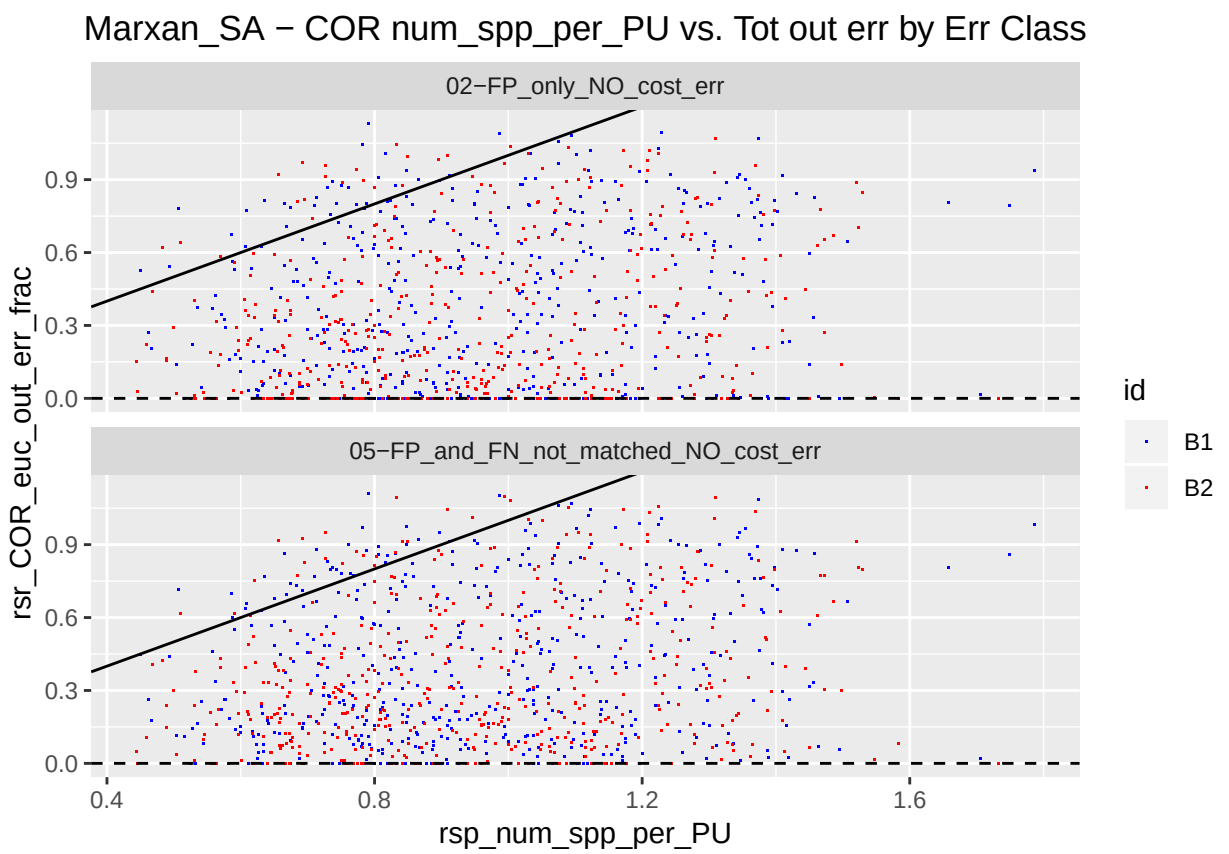
```
gg_rsp_num_spp_vs_eucOutTot_facetted_by_FP_dominance (rs_name,
                                                      FP_dominant_working_df,
                                                      ref_y = 0)
```




```
#================================================================================

gg_rsp_num_spp_per_PU_vs_eucOutTot_facetted_by_FP_dominance <- function (rs_name,
                                                                         sorted_msa_tib,
                                                                         ref_y = 0
                                                                         , shape_type = "."
                                                                         , alpha_level = 1
                                                                         )
    {
    ggplot (data = sorted_msa_tib) +
      geom_point (mapping = aes(x = rsp_num_spp_per_PU,
                                y = rsr_COR_euc_out_err_frac,
    #                            color = rsp_combined_err_label)) +
    #                             color = id
    # ),
    # shape="."
    # ) +
 # color = id
 #    ),
 #    shape="."
 #    ) +
 color = id
    ),
    shape="."
    ) +
```

```
    #scale_color_viridis(discrete=TRUE) +
    #scale_color_brewer(palette="Set1") +
    scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

    ggtitle (paste0 (rs_name, " - COR num_spp_per_PU vs. Tot out err by Err Class")) +
    theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

      facet_wrap (~ rsp_combined_err_label, nrow = 2) +
      geom_hline (yintercept = ref_y, linetype="dashed",
                  color = "black", size=0.5) +
      geom_abline (intercept=0, slope=1  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=5  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=10  #, linetype, color, size
                   )
    }
```

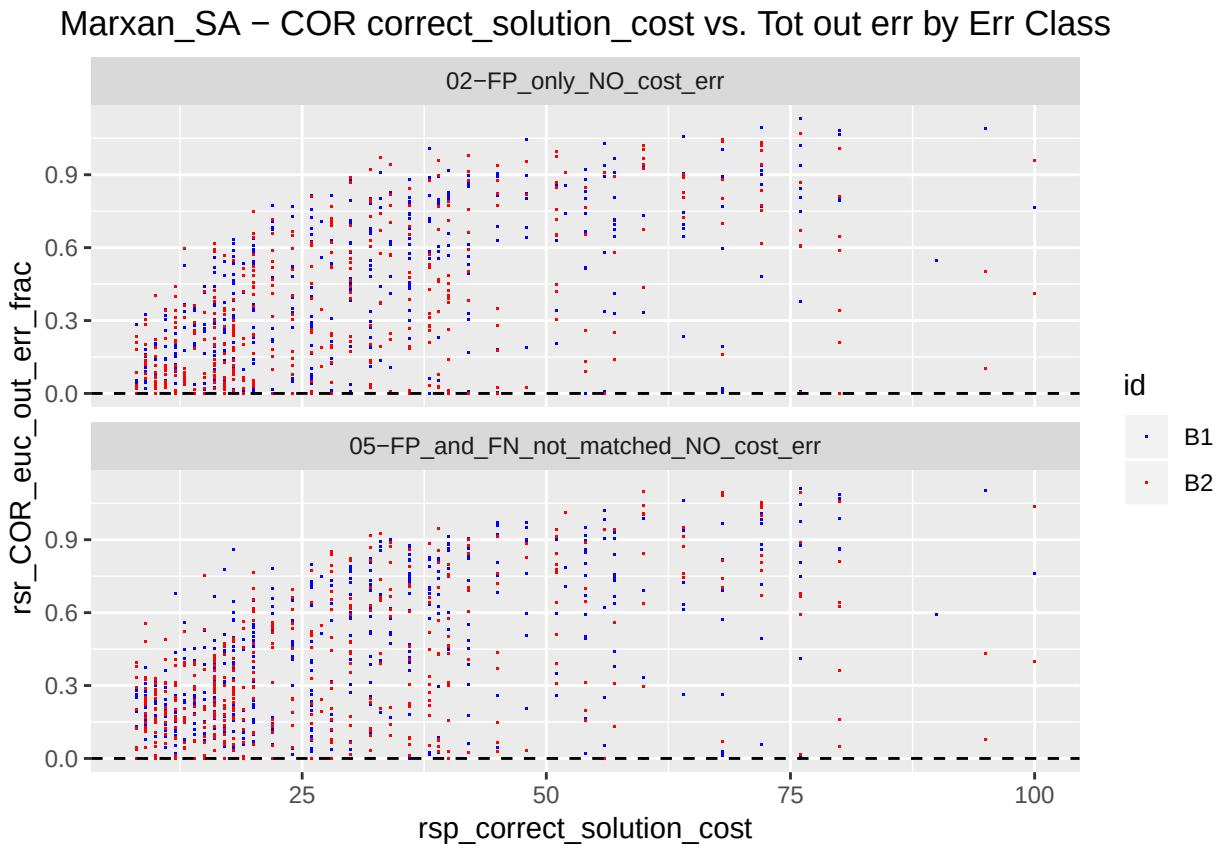
```
gg_rsp_num_spp_per_PU_vs_eucOutTot_facetted_by_FP_dominance (rs_name,
                                                             FP_dominant_working_df,
                                                             ref_y = 0)
```




```
#=============================================================================

gg_rsp_correct_solution_cost_vs_eucOutTot_facetted_by_FP_dominance <- function (rs_name,
                                                                  sorted_msa_tib,
                                                                  ref_y = 0
```

```
                                                        , shape_type = "."
                                                        , alpha_level = 1
                                                        )
    {
    ggplot (data = sorted_msa_tib) +
      geom_point (mapping = aes(x = rsp_correct_solution_cost,
                                y = rsr_COR_euc_out_err_frac,
    #                            color = rsp_combined_err_label)) +
    #                             color = id
    # ),
    # shape="."
    # ) +
 # color = id
 #    ),
 #    shape="."
 #    ) +
 color = id
    ),
    shape="."
    ) +


    #scale_color_viridis(discrete=TRUE) +
    #scale_color_brewer(palette="Set1") +
    scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

    ggtitle (paste0 (rs_name, " - COR correct_solution_cost vs. Tot out err by Err Class")) +
    theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

      facet_wrap (~ rsp_combined_err_label, nrow = 2) +
      geom_hline (yintercept = ref_y, linetype="dashed",
                  color = "black", size=0.5) +
      geom_abline (intercept=0, slope=1  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=5  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=10  #, linetype, color, size
                   )
    }
```

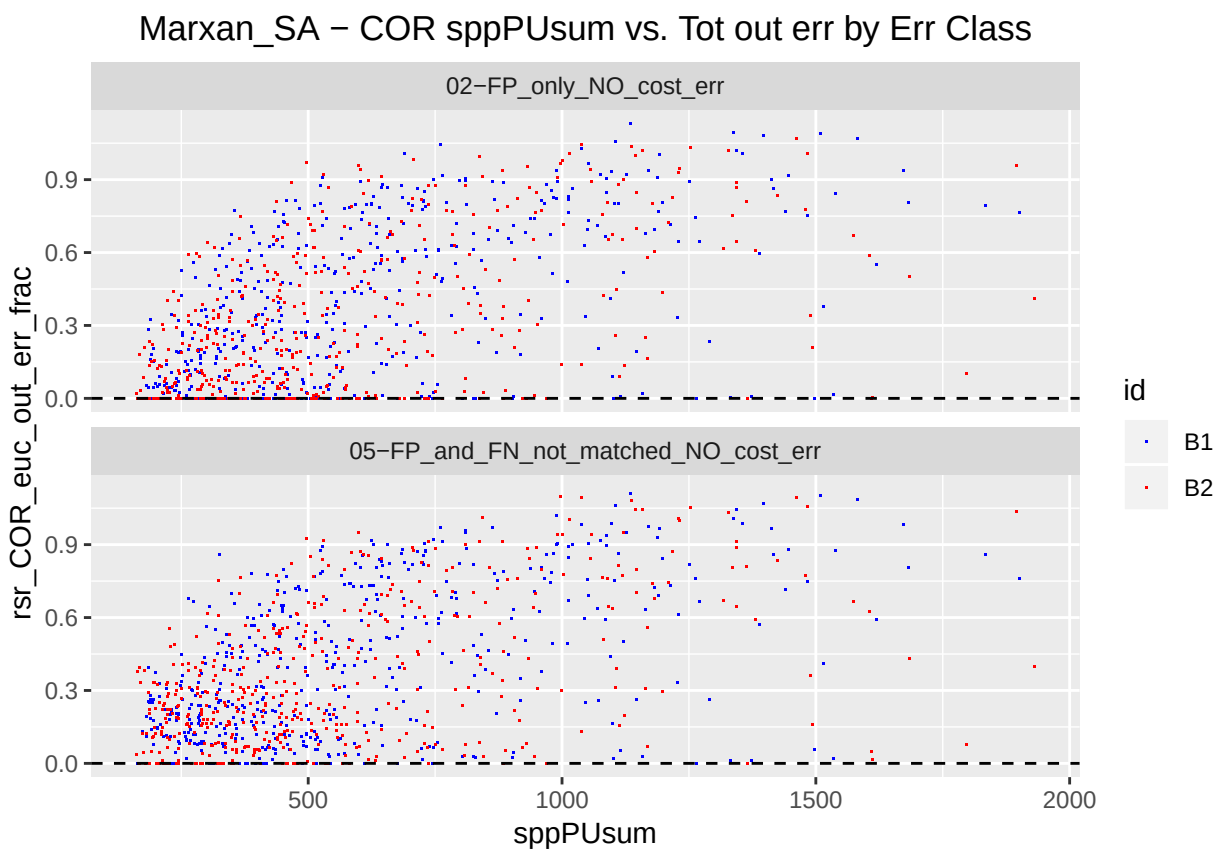
```
gg_rsp_correct_solution_cost_vs_eucOutTot_facetted_by_FP_dominance (rs_name,
                                                    FP_dominant_working_df,
                                                    ref_y = 0)
```




```
#==============================================================================

gg_sppPUsum_vs_eucOutTot_facetted_by_FP_dominance <- function (rs_name,
                                                               sorted_msa_tib,
                                                               ref_y = 0
                                                               , shape_type = "."
                                                               , alpha_level = 1
                                                               )
    {
    ggplot (data = sorted_msa_tib) +
      geom_point (mapping = aes(x = sppPUsum,
                                y = rsr_COR_euc_out_err_frac,
    #                            color = rsp_combined_err_label)) +
    #                             color = id
    # ),
    # shape="."
    # ) +
 # color = id
 #    ),
 #    shape="."
 #    ) +
 color = id
    ),
    shape="."
    ) +
```

```
    #scale_color_viridis(discrete=TRUE) +
    #scale_color_brewer(palette="Set1") +
    scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

    ggtitle (paste0 (rs_name, " - COR sppPUsum vs. Tot out err by Err Class")) +
    theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

      facet_wrap (~ rsp_combined_err_label, nrow = 2) +
      geom_hline (yintercept = ref_y, linetype="dashed",
                  color = "black", size=0.5) +
      geom_abline (intercept=0, slope=1  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=5  #, linetype, color, size
                   ) +
      geom_abline (intercept=0, slope=10  #, linetype, color, size
                   )
    }
```

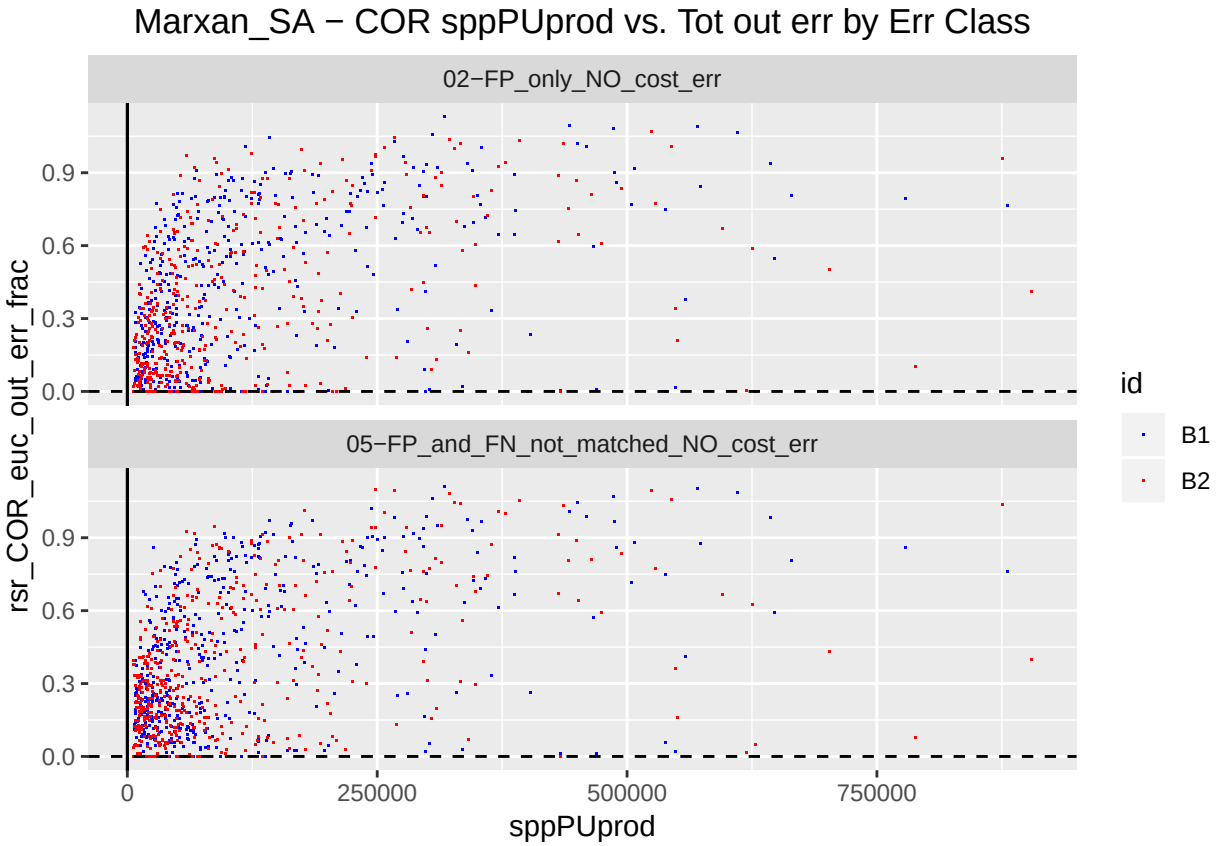
```
gg_sppPUsum_vs_eucOutTot_facetted_by_FP_dominance (rs_name,
                                                   FP_dominant_working_df,
                                                   ref_y = 0)
```




```
#=============================================================================

gg_sppPUprod_vs_eucOutTot_facetted_by_FP_dominance <- function (rs_name,
                                                                sorted_msa_tib,
                                                                ref_y = 0
```

```
                                                          , shape_type = "."
                                                          , alpha_level = 1
                                                          )
   {
   ggplot (data = sorted_msa_tib) +
     geom_point (mapping = aes(x = sppPUprod,
                               y = rsr_COR_euc_out_err_frac,
   #                            color = rsp_combined_err_label)) +
   #                             color = id
   # ),
   # shape="."
   # ) +
# color = id
#    ),
#    shape="."
#    ) +
color = id
   ),
   shape="."
   ) +


   #scale_color_viridis(discrete=TRUE) +
   #scale_color_brewer(palette="Set1") +
   scale_color_manual(breaks = c("B1", "B2"), values=c("blue", "red")) +

   ggtitle (paste0 (rs_name, " - COR sppPUprod vs. Tot out err by Err Class")) +
   theme(plot.title = element_text(hjust = 0.5)) +    #  To center the title

     facet_wrap (~ rsp_combined_err_label, nrow = 2) +
     geom_hline (yintercept = ref_y, linetype="dashed",
                 color = "black", size=0.5) +
     geom_abline (intercept=0, slope=1  #, linetype, color, size
                  ) +
     geom_abline (intercept=0, slope=5  #, linetype, color, size
                  ) +
     geom_abline (intercept=0, slope=10  #, linetype, color, size
                  )
   }
```

```
gg_sppPUprod_vs_eucOutTot_facetted_by_FP_dominance (rs_name,
                                                    FP_dominant_working_df,
                                                    ref_y = 0)
```



7.5 Different types of output error

7.5.1 Ascelin's comment about how much effort is each RS putting into getting the tail end of the species

- I think that I do have a kind of record of that in the cumulative plots I was putting out showing fraction of species covered on x and number of patches included (cost) on y. I think those plots are still being generated. Need to look. It may be that some of them have an elbow that could be identified, but if there's no elbow, the choice of cutoff becomes more arbitrary.

- Other possibilities might be:
  - fraction of species covered by only spending the optimal cost (i.e., consider that the budget)
  - fraction of species covered at various quantiles of cost (e.g., 90, 95, 99)
  - fraction of optimal cost for various quantiles of fraction of spp covered (e.g., 90, 95, 99)
  - or just a simple invocation of the 80/20 rule in some way
    - I think that at one point, I was calculating the cost per spp covered at each step in the purchase sequence, but the maximum value of that occurs at the start of the process, so it's not that helpful

    - then again, you could start throwing out PUs based on unprotected richness or something until you get down to the budget

- One issue here is that when the non-greedy reserve selectors are running on the apparent data, they don't give you an ordering of the PUs and they're making mistakes *because* they don't know they're making them, so you can only do some sort of correct pruning if you do know the correct answer.

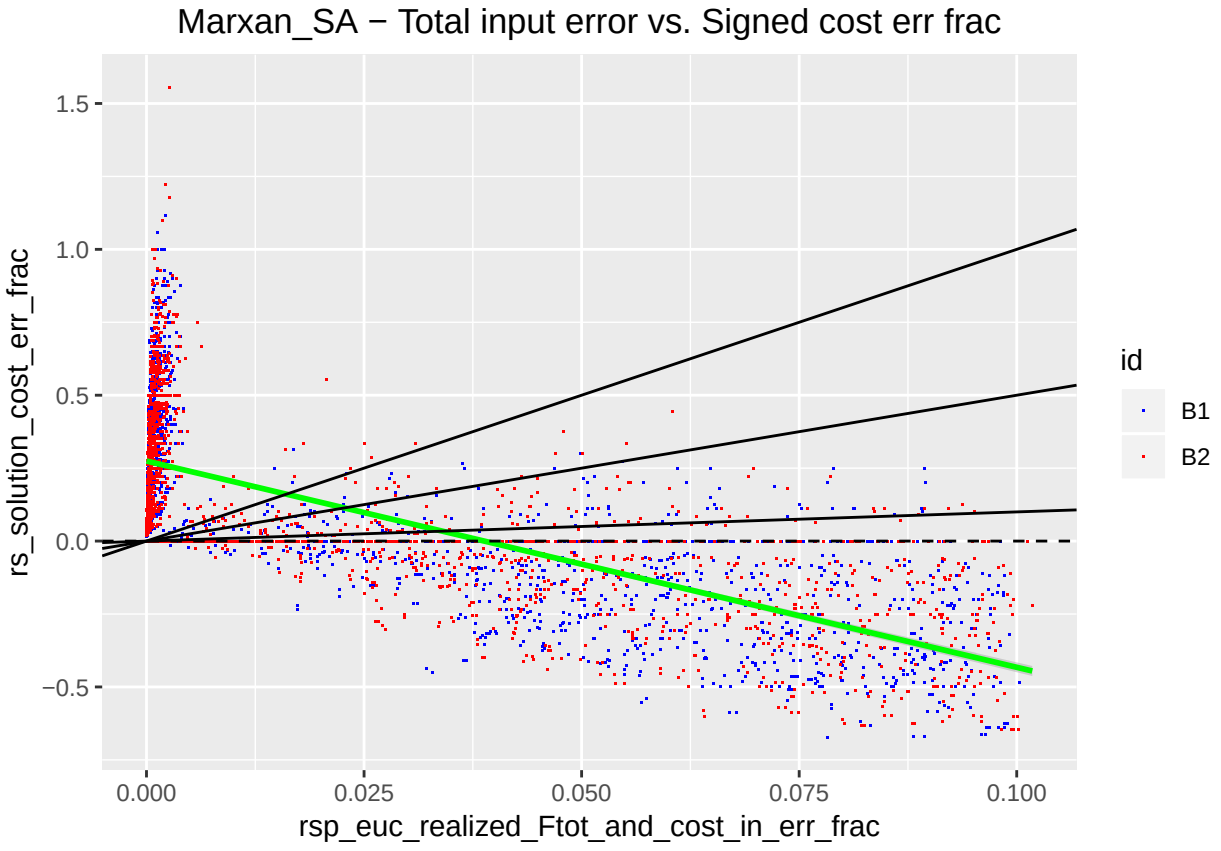
7.5.2 Decompose output errors into over/under, etc

I've tracked the overcost and undercost and the underrepresentation (shortfall). Maybe these can partly get at what Ascelin was asking about.

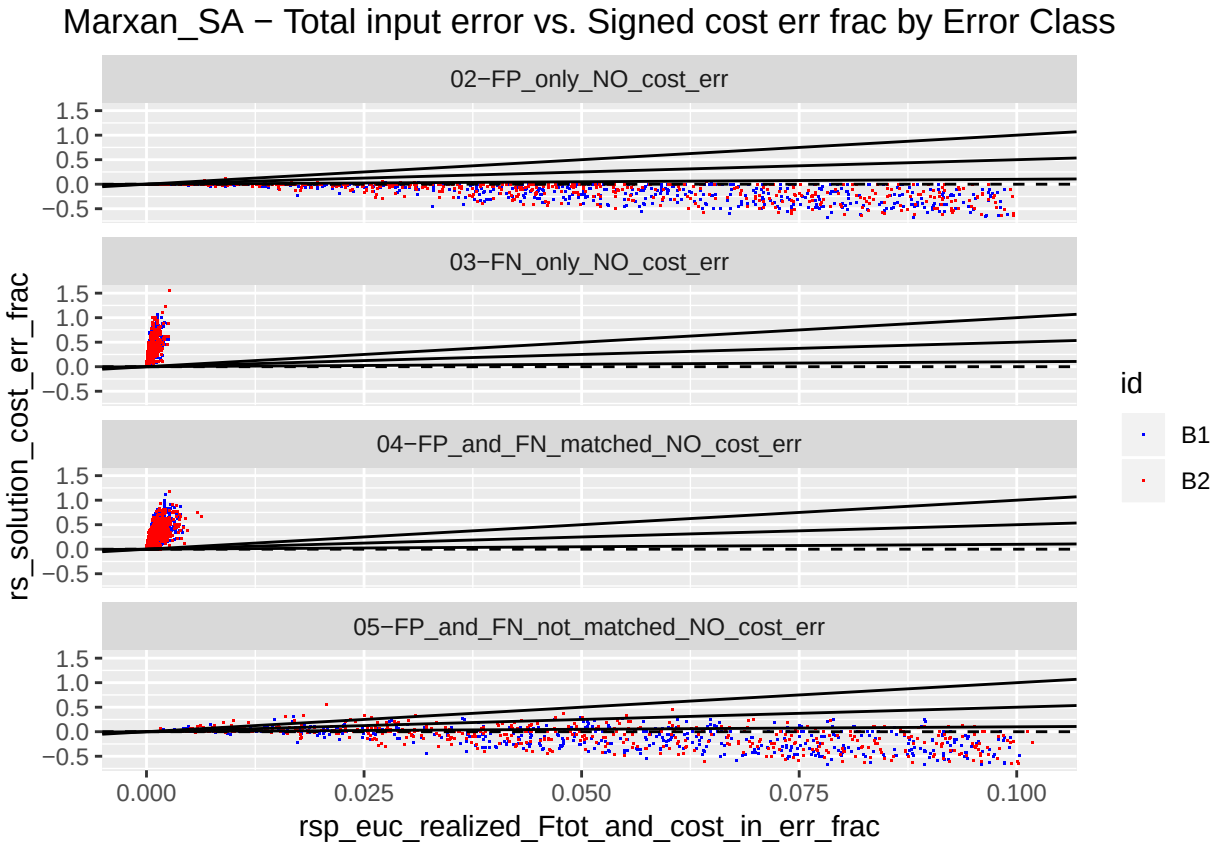
7.5.2.1 Results and their errors

- rs_solution_cost_err_frac
- abs_rs_solution_cost_err_frac
- rsr_COR_spp_rep_shortfall
- rsr_COR_solution_FRAC_spp_covered

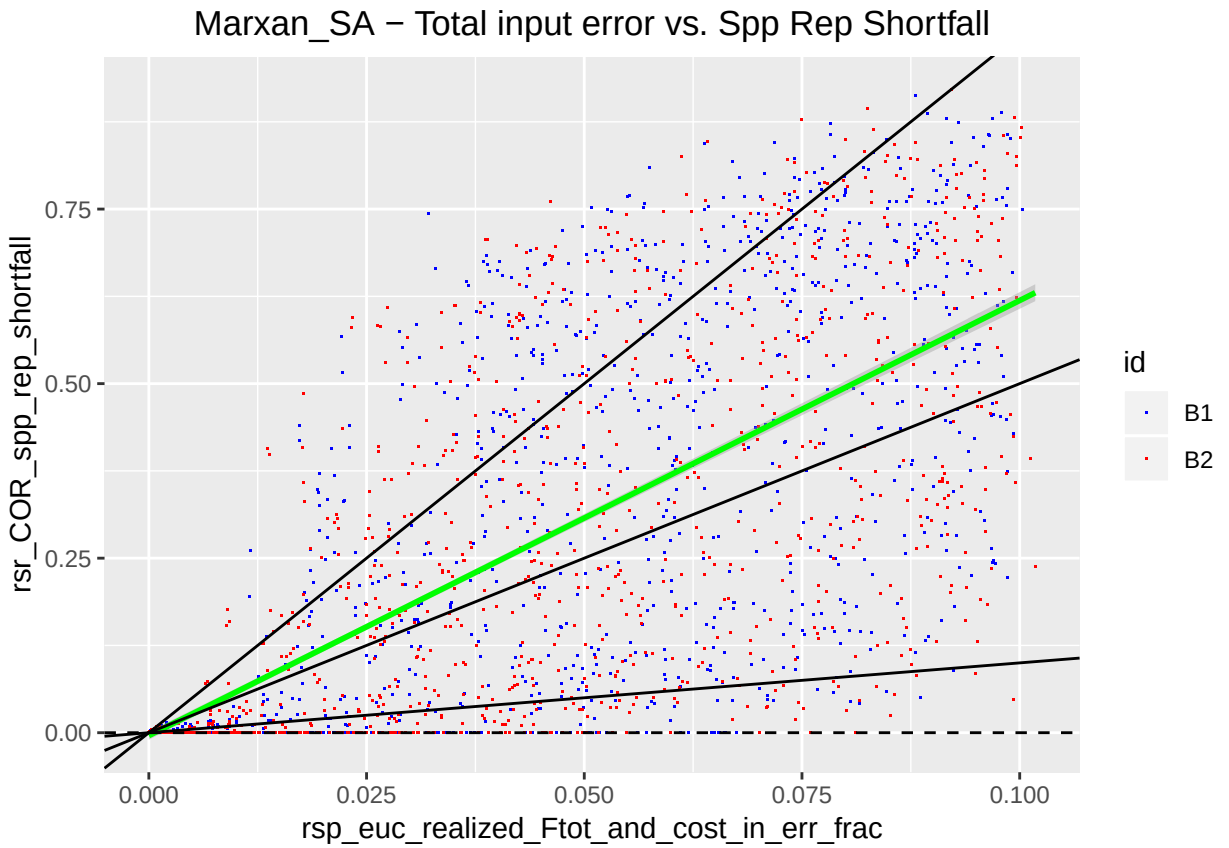
```
#--  rs_solution_cost_err_frac

#####gg_eucInTot_vs_signedCostErrFrac_all_on_1 (rs_name, easyHard_df, ref_y = 0)
#####gg_eucInTot_vs_signedCostErrFrac_facetted_by_err_label (rs_name, easyHard_df, ref_y = 0)
# easyHard_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_eucInTot_vs_signedCostErrFrac_all_on_1 (rs_name, easyHard_not_huge_out_err_df, ref_y = 0)
#####gg_eucInTot_vs_signedCostErrFrac_facetted_by_err_label (rs_name, easyHard_not_huge_out_err_df, ref_

gg_eucInTot_vs_signedCostErrFrac_all_on_1 (rs_name, working_df, ref_y = 0)
```



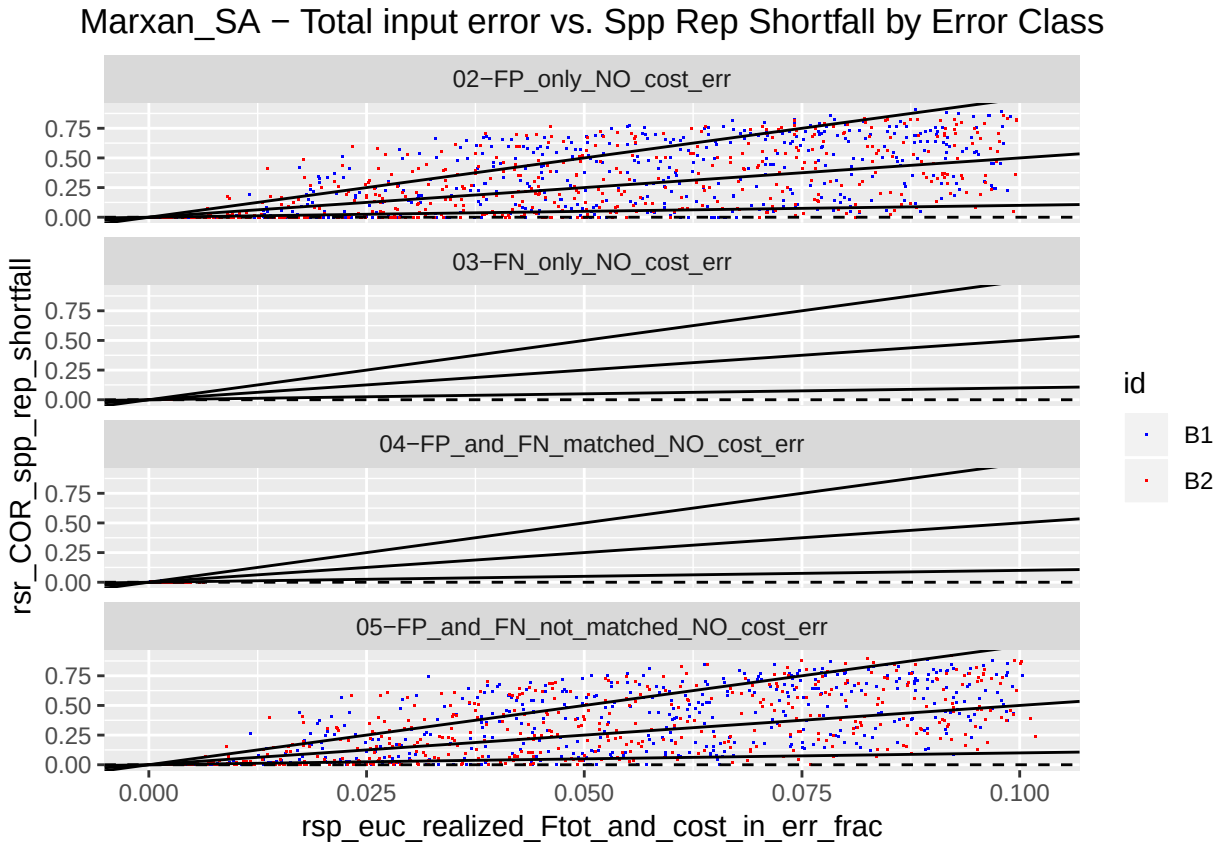

```
gg_eucInTot_vs_signedCostErrFrac_facetted_by_err_label (rs_name, working_df, ref_y = 0)
```




```
# working_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_eucInTot_vs_signedCostErrFrac_all_on_1 (rs_name, working_not_huge_out_err_df, ref_y = 0)
#####gg_eucInTot_vs_signedCostErrFrac_facetted_by_err_label (rs_name, working_not_huge_out_err_df, ref_y

#--  rsr_COR_spp_rep_shortfall

#####gg_eucInTot_vs_sppRepShortfall_all_on_1 (rs_name, easyHard_df, ref_y = 0)
#####gg_eucInTot_vs_sppRepShortfall_facetted_by_err_label (rs_name, easyHard_df, ref_y = 0)
# easyHard_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_eucInTot_vs_sppRepShortfall_all_on_1 (rs_name, easyHard_not_huge_out_err_df, ref_y = 0)
#####gg_eucInTot_vs_sppRepShortfall_facetted_by_err_label (rs_name, easyHard_not_huge_out_err_df, ref_y

gg_eucInTot_vs_sppRepShortfall_all_on_1 (rs_name, working_df, ref_y = 0)
```



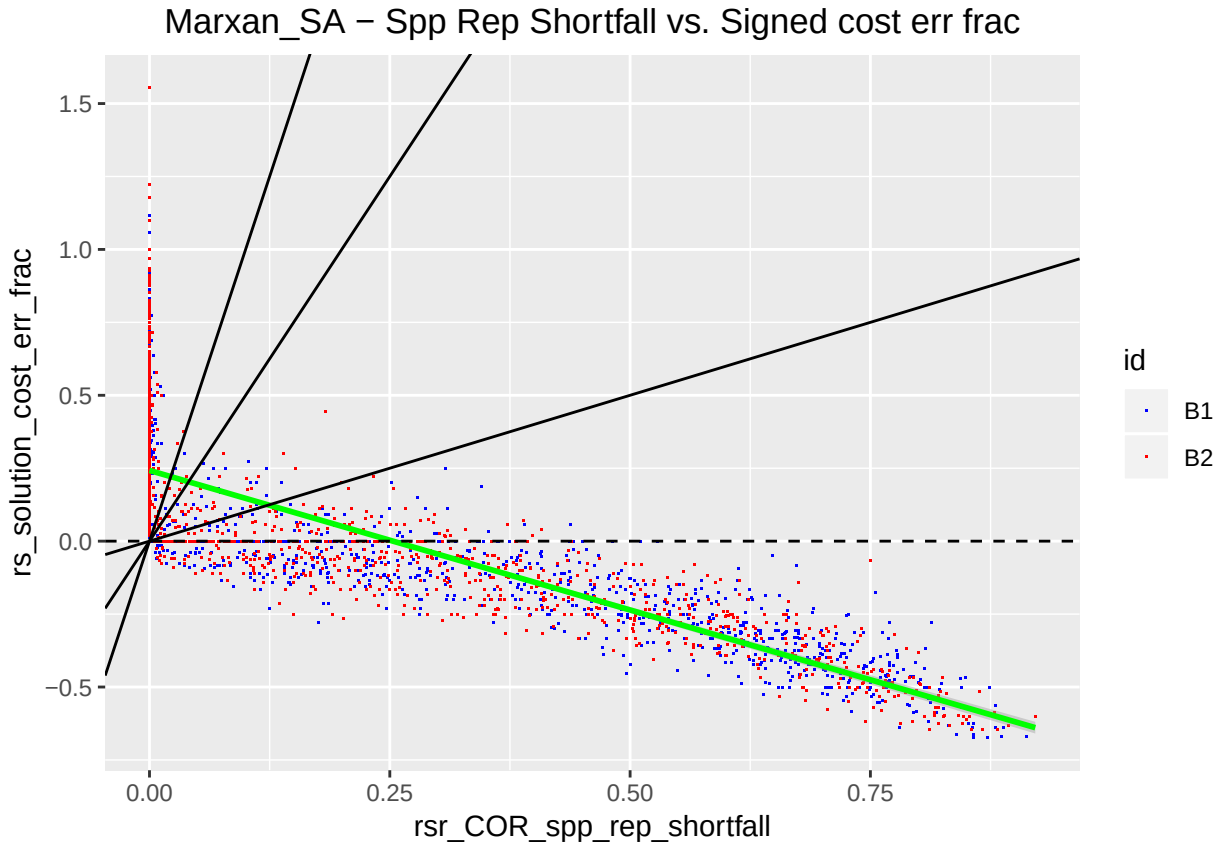

```
gg_eucInTot_vs_sppRepShortfall_facetted_by_err_label (rs_name, working_df, ref_y = 0)
```



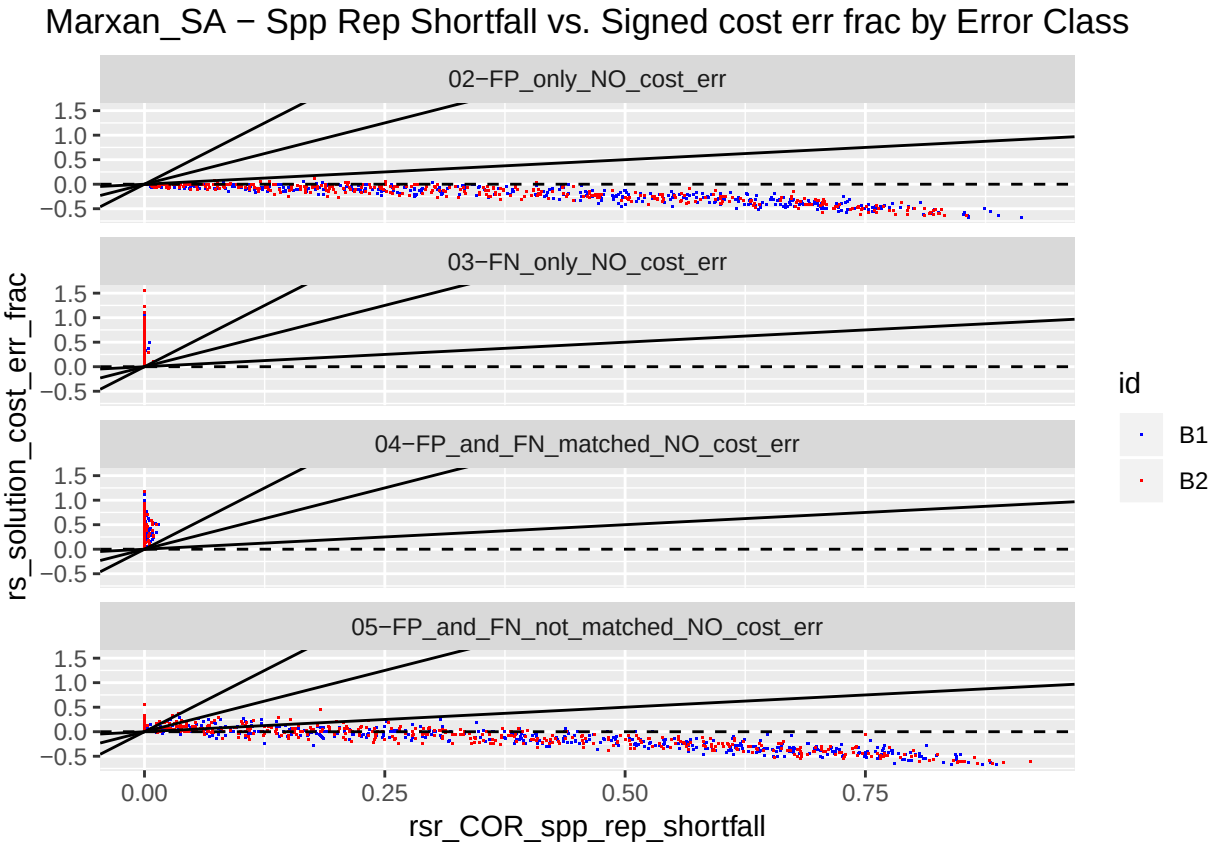

```
# working_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> working_not_huge_out_err_df
#####gg_eucInTot_vs_sppRepShortfall_all_on_1 (rs_name, working_not_huge_out_err_df, ref_y = 0)
#####gg_eucInTot_vs_sppRepShortfall_facetted_by_err_label (rs_name, working_not_huge_out_err_df, ref_y =
```

7.5.2.2 Spp rep shortfall vs. Signed cost error

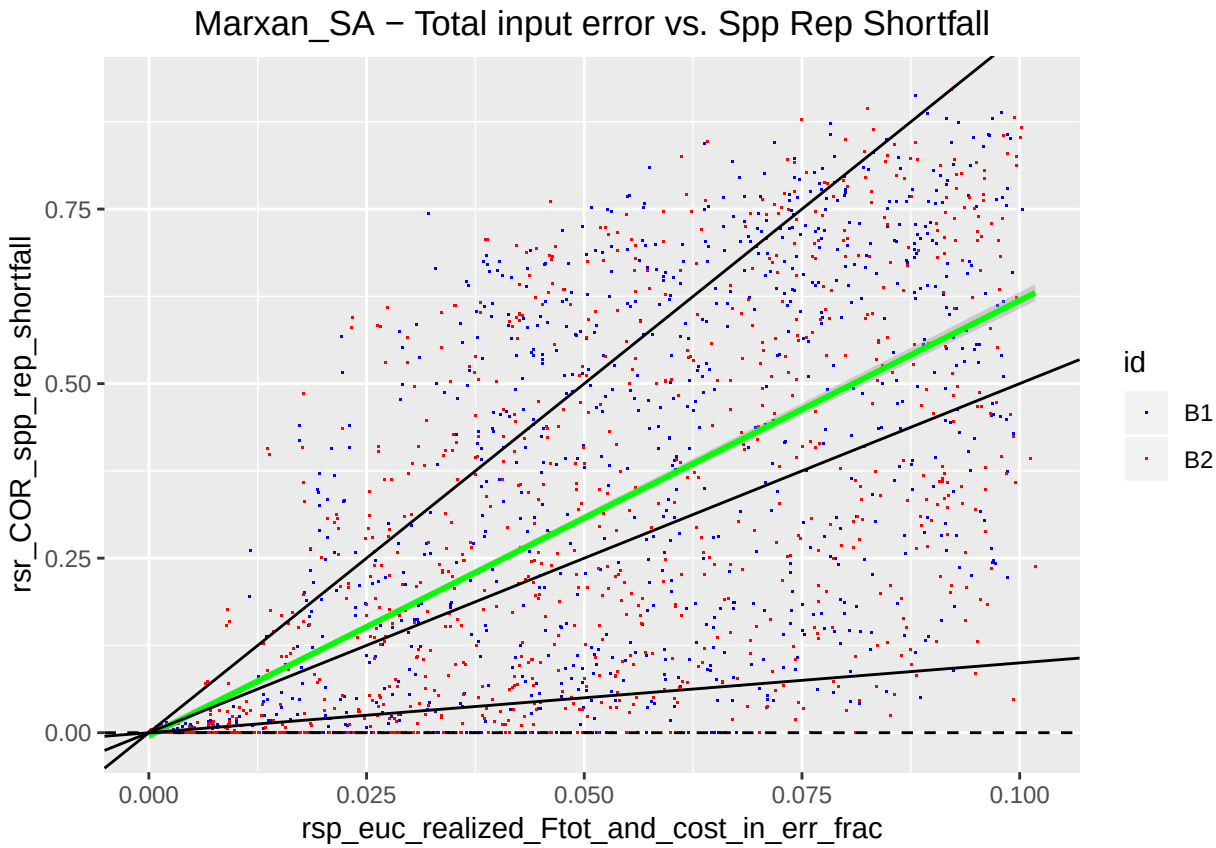
```
#--  rs_solution_cost_err_frac

#####gg_sppRepShortfall_vs_signedCostErrFrac_all_on_1 (rs_name, easyHard_df, ref_y = 0)
#####gg_sppRepShortfall_vs_signedCostErrFrac_facetted_by_err_label (rs_name, easyHard_df, ref_y = 0)
# easyHard_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_sppRepShortfall_vs_signedCostErrFrac_all_on_1 (rs_name, easyHard_not_huge_out_err_df, ref_y = 0
#####gg_sppRepShortfall_vs_signedCostErrFrac_facetted_by_err_label (rs_name, easyHard_not_huge_out_err_

gg_sppRepShortfall_vs_signedCostErrFrac_all_on_1 (rs_name, working_df, ref_y = 0)
```



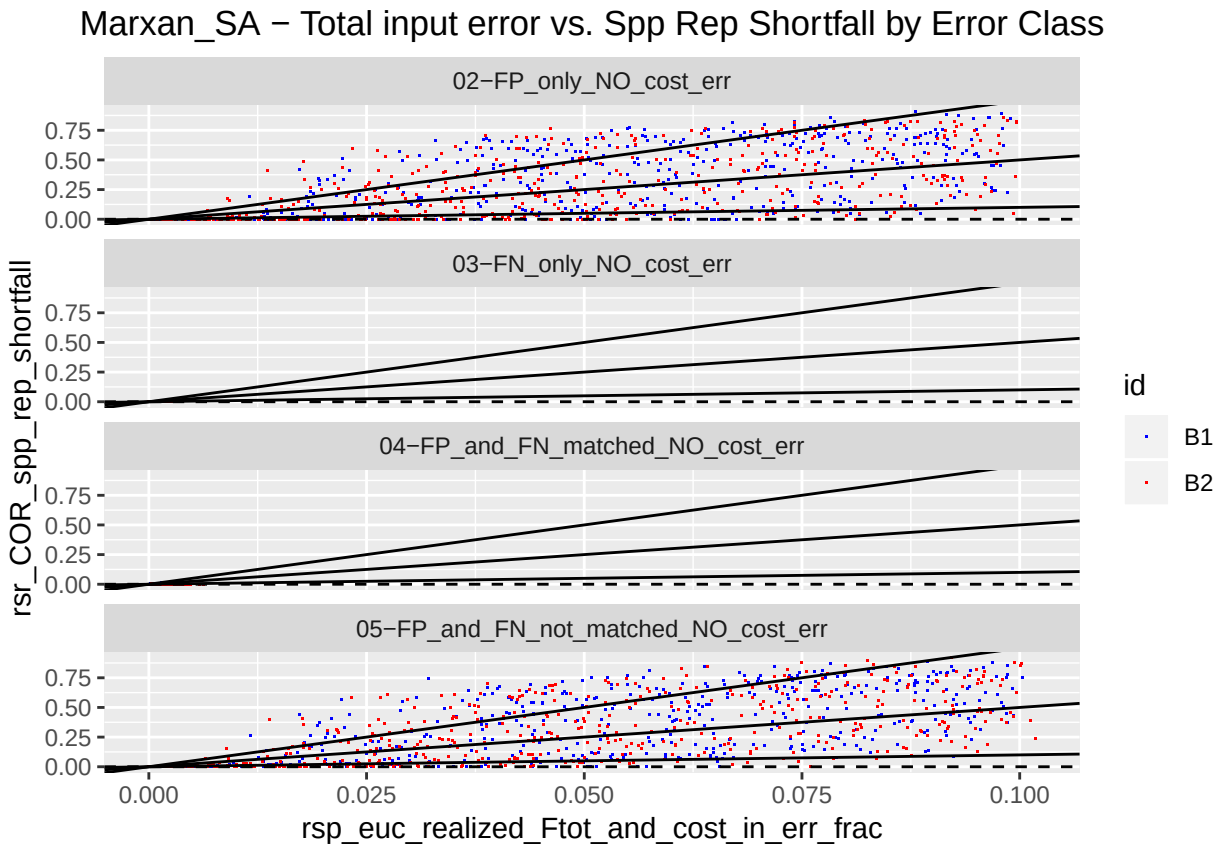

```
gg_sppRepShortfall_vs_signedCostErrFrac_facetted_by_err_label (rs_name, working_df, ref_y = 0)
```




```
# working_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_sppRepShortfall_vs_signedCostErrFrac_all_on_1 (rs_name, working_not_huge_out_err_df, ref_y = 0)
#####gg_sppRepShortfall_vs_signedCostErrFrac_facetted_by_err_label (rs_name, working_not_huge_out_err_d

#--  rsr_COR_spp_rep_shortfall

#####gg_eucInTot_vs_sppRepShortfall_all_on_1 (rs_name, easyHard_df, ref_y = 0)
#####gg_eucInTot_vs_sppRepShortfall_facetted_by_err_label (rs_name, easyHard_df, ref_y = 0)
# easyHard_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_eucInTot_vs_sppRepShortfall_all_on_1 (rs_name, easyHard_not_huge_out_err_df, ref_y = 0)
#####gg_eucInTot_vs_sppRepShortfall_facetted_by_err_label (rs_name, easyHard_not_huge_out_err_df, ref_y

gg_eucInTot_vs_sppRepShortfall_all_on_1 (rs_name, working_df, ref_y = 0)
```




```
gg_eucInTot_vs_sppRepShortfall_facetted_by_err_label (rs_name, working_df, ref_y = 0)
```




```
# working_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_eucInTot_vs_sppRepShortfall_all_on_1 (rs_name, working_not_huge_out_err_df, ref_y = 0)
#####gg_eucInTot_vs_sppRepShortfall_facetted_by_err_label (rs_name, working_not_huge_out_err_df, ref_y =
```

7.5.2.3 Also look at problem of putting errors on equal footing, i.e., how much error is *possible*

I've also tracked the *fraction of possible* overcost and undercost as a way of trying to put problems on a more equal footing.

- Overcost:
  - The largest possible error is related to what fraction of the landscape is required in the optimal solution. If the optimal solution takes up most of the landscape, then only a tiny overcost is possible.
  - max overcost error = (total landscape cost - optimal cost) / optimal cost
  - `= (total landscape cost / optimal cost) - 1`
- Undercost:
  - In all cases, the undercost is bounded by 100%, since the error is a fraction of the optimal cost and costs can't go less than 0 and once you hit 0 cost error is 100%.
  - max undercost error = (optimal cost - min cost possible) / optimal cost

– = (optimal cost - 0) / optimal cost = 1

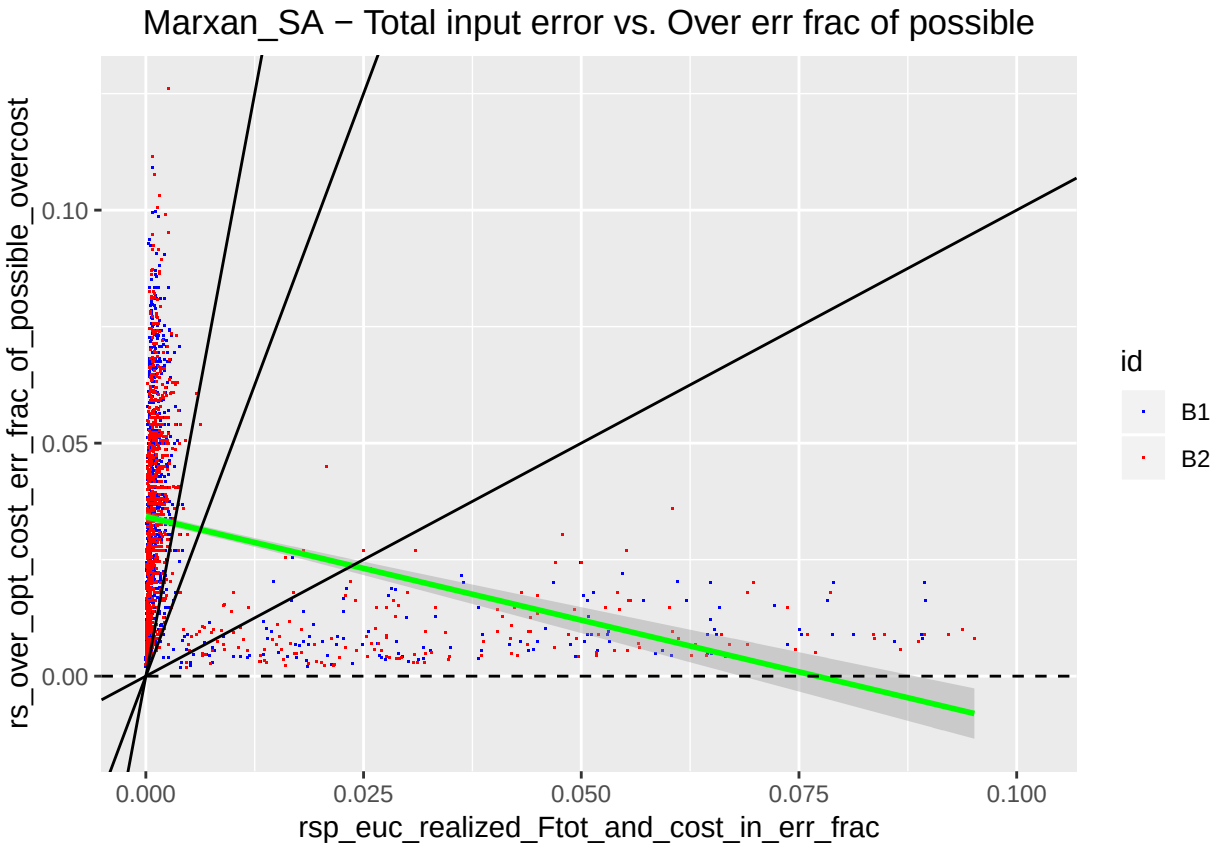
7.5.2.4 Results and their errors

- rs_over_opt_cost_err_frac_of_possible_overcost
- rs_under_opt_cost_err_frac_of_possible_undercost

```
#--  OVER cost frac possible

#####gg_eucInTot_vs_fracPossibleOver_all_on_1 (rs_name, easyHard_df, ref_y = 0)
#####gg_eucInTot_vs_fracPossibleOver_facetted_by_err_label (rs_name, easyHard_df, ref_y = 0)
# easyHard_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_eucInTot_vs_fracPossibleOver_all_on_1 (rs_name, easyHard_not_huge_out_err_df, ref_y = 0)
#####gg_eucInTot_vs_fracPossibleOver_facetted_by_err_label (rs_name, easyHard_not_huge_out_err_df, ref_y

gg_eucInTot_vs_fracPossibleOver_all_on_1 (rs_name, working_df, ref_y = 0)
```

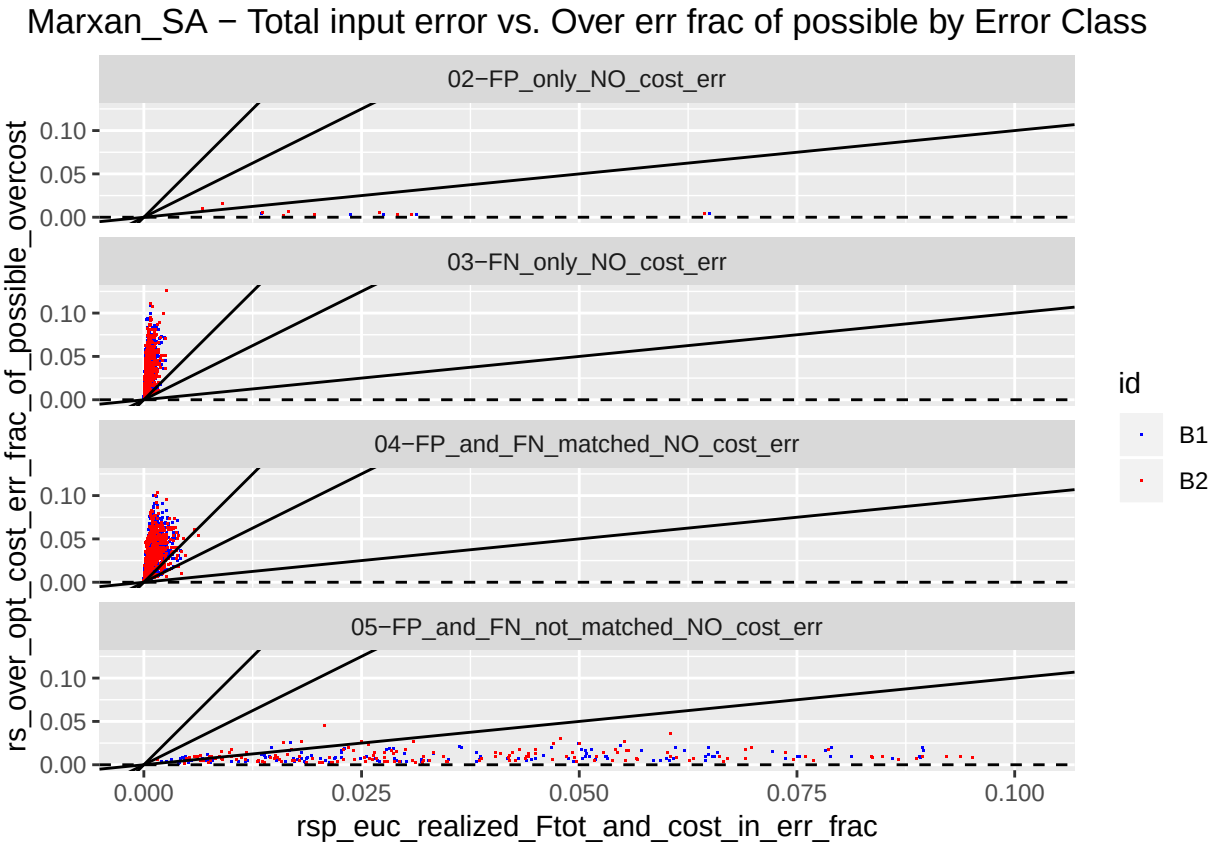
```
## Warning: Removed 1820 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1820 rows containing missing values (geom_point).
```




```
gg_eucInTot_vs_fracPossibleOver_facetted_by_err_label (rs_name, working_df, ref_y = 0)
```

```
## Warning: Removed 1820 rows containing missing values (geom_point).
```



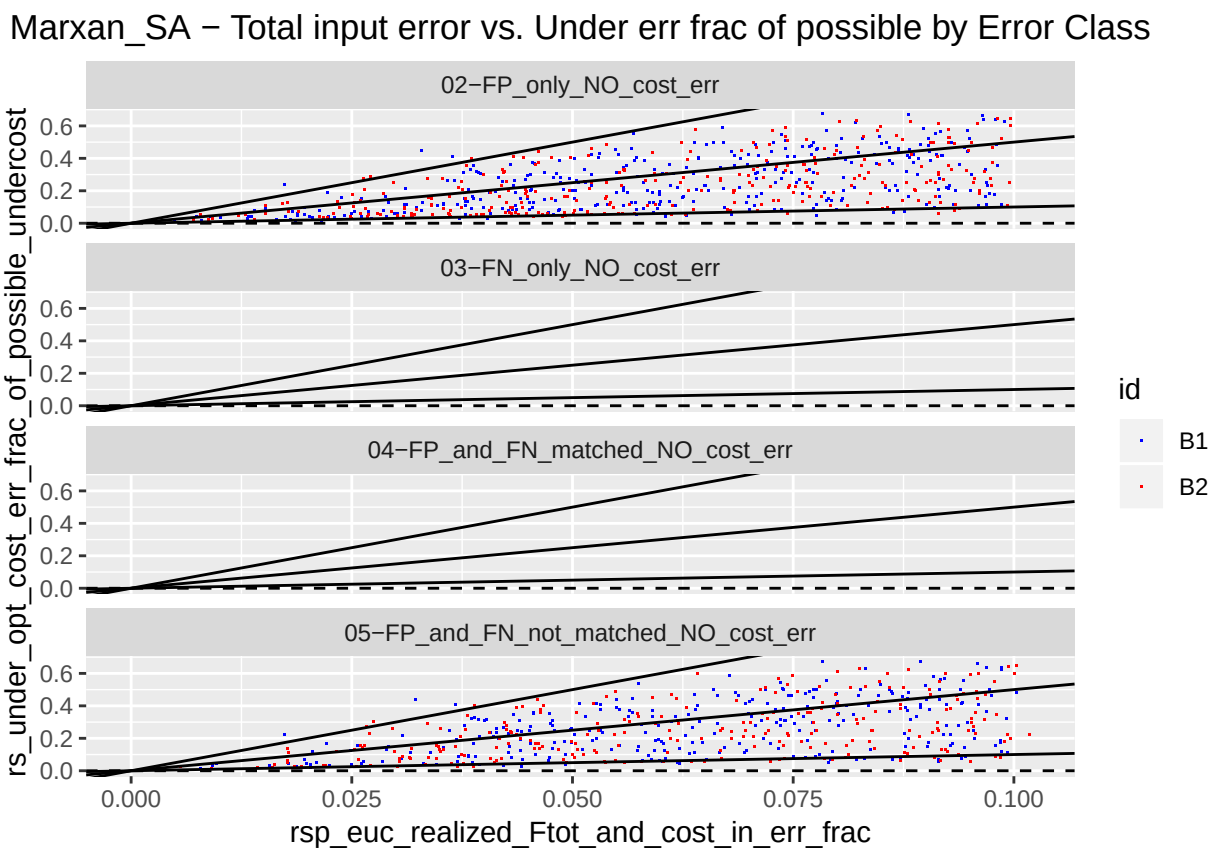

```
# working_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_eucInTot_vs_fracPossibleOver_all_on_1 (rs_name, working_not_huge_out_err_df, ref_y = 0)
#####gg_eucInTot_vs_fracPossibleOver_facetted_by_err_label (rs_name, working_not_huge_out_err_df, ref_y

#--  UNDER cost frac possible

#####gg_eucInTot_vs_fracPossibleUnder_all_on_1 (rs_name, easyHard_df, ref_y = 0)
#####gg_eucInTot_vs_fracPossibleUnder_facetted_by_err_label (rs_name, easyHard_df, ref_y = 0)
# easyHard_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_eucInTot_vs_fracPossibleUnder_all_on_1 (rs_name, easyHard_not_huge_out_err_df, ref_y = 0)
#####gg_eucInTot_vs_fracPossibleUnder_facetted_by_err_label (rs_name, easyHard_not_huge_out_err_df, ref

gg_eucInTot_vs_fracPossibleUnder_all_on_1 (rs_name, working_df, ref_y = 0)
```

```
## Warning: Removed 2810 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 2810 rows containing missing values (geom_point).
```


```
gg_eucInTot_vs_fracPossibleUnder_facetted_by_err_label (rs_name, working_df, ref_y = 0)
```

```
## Warning: Removed 2810 rows containing missing values (geom_point).
```




```
# working_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df
#####gg_eucInTot_vs_fracPossibleUnder_all_on_1 (rs_name, working_not_huge_out_err_df, ref_y = 0)
#####gg_eucInTot_vs_fracPossibleUnder_facetted_by_err_label (rs_name, working_not_huge_out_err_df, ref_y
```

---

8 Predicting performance of specific RS on specific problems

Given all this variability, can you predict how a particular reserve selector will perform on a specific problem?

8.1 Compute correlations among primary working data variables

```
library (corrplot)
```

```
## corrplot 0.84 loaded
```

8.1.1 Old version before timings and bipartite measures

```
    #  Compute correlation matrix after stripping out non-continuous
    #  variables.
continuous_vars_working_df =
```

```
select (working_df, #c (

                #  Input descriptors
                rsp_num_PUs,
                rsp_num_spp,
                rsp_num_spp_per_PU,
                rsp_correct_solution_cost,

                sppPUsum,
                sppPUprod,

        #  Post-gen knowable problem descriptors

            #  Species and PU counts
                    # rsp_app_num_spp,
                    # rsp_app_num_PUs,

            #  igraph package metrics
#                ig_rsp_UUID,

                    # ig_top,
                    # ig_bottom,
                    # ig_num_edges_m,
                    # ig_ktop,
                    # ig_kbottom,
                    # ig_bidens,
                    # ig_lcctop,
                    # ig_lccbottom,
                    # ig_distop,
                    # ig_disbottom,
                    # ig_cctop,
                    # ig_ccbottom,
                    # ig_cclowdottop,
                    # ig_cclowdotbottom,
                    # ig_cctopdottop,
                    # ig_cctopdotbottom,
                ig_mean_bottom_bg_redundancy,
                ig_median_bottom_bg_redundancy,
                ig_mean_top_bg_redundancy,
                ig_median_top_bg_redundancy,

                # ig_user_time,
                # ig_system_time,
                # ig_elapsed_time,
                # ig_user_child_time,
                # ig_sys_child_time,

            #  bipartite package metrics
#                bip_rsp_UUID,

                connectance,
                        # number.of.PUs,
                        # number.of.Spp,
```

```
                    # bip_user_time,
                    # bip_system_time,
                    # bip_elapsed_time,
                    # bip_user_child_time,
                    # bip_sys_child_time,


                #  Possibly knowable realized input error values
                            # rsp_realized_median_abs_cost_err_frac,
                            # rsp_realized_mean_abs_cost_err_frac,
                            # rsp_realized_sd_abs_cost_err_frac,
                            # rsp_FP_const_rate,
                            # rsp_FN_const_rate,

                    rsp_realized_FP_rate,
                    rsp_realized_FN_rate,
                    rsp_realized_Ftot_rate,
                    rsp_euc_realized_FP_and_cost_in_err_frac,
                    rsp_euc_realized_FN_and_cost_in_err_frac,
                    rsp_euc_realized_Ftot_and_cost_in_err_frac,

#                       rsp_wrap_is_imperfect,


                #  Results and their errors
                            # rs_solution_cost,
                            # rs_solution_cost_err_frac,
                            # abs_rs_solution_cost_err_frac,
                            #
                            # rs_over_opt_cost_err_frac_of_possible_overcost,
                            # rs_under_opt_cost_err_frac_of_possible_undercost,
                            #
                    rsr_COR_euc_out_err_frac,
                            #
                            # rsr_COR_spp_rep_shortfall,
                            # rsr_COR_solution_NUM_spp_covered,
                            # rsr_COR_solution_FRAC_spp_covered,

                    err_mag
    )
#)
```

```
correlations = cor (continuous_vars_working_df)

dim (correlations)
```

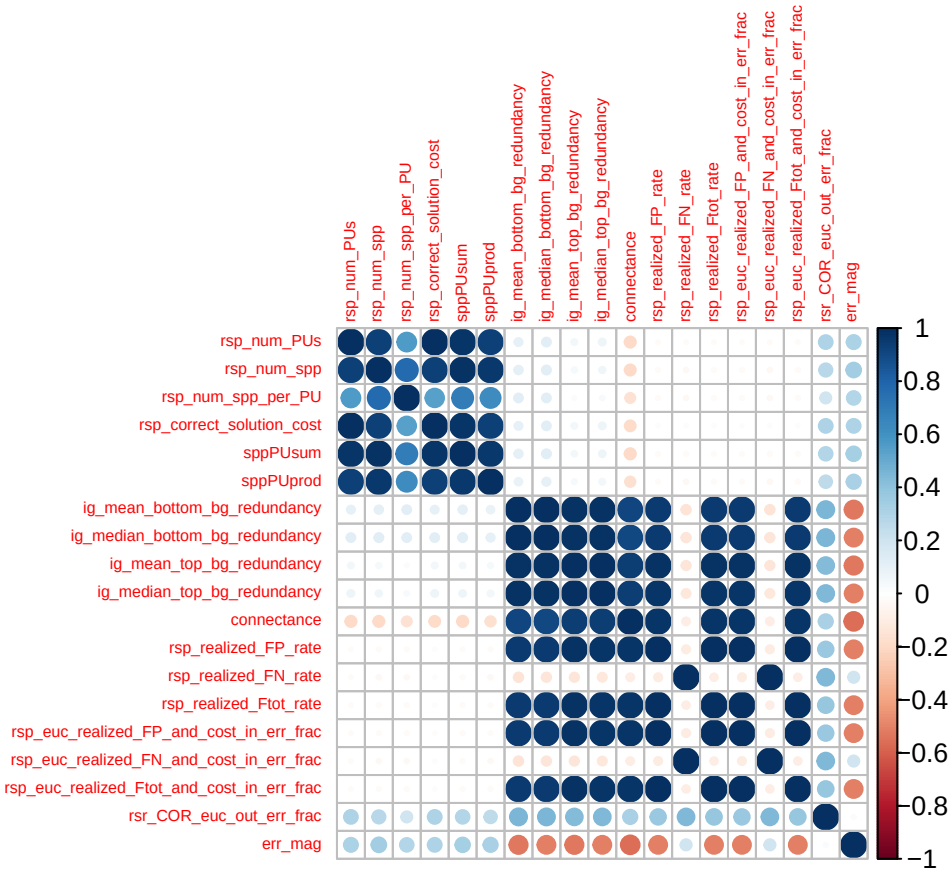
```
## [1] 19 19
```

```
correlations [, 18:19]
```

```
##                                          rsr_COR_euc_out_err_frac
## rsp_num_PUs                                           0.30165423
## rsp_num_spp                                           0.27871255
## rsp_num_spp_per_PU                                    0.19602256
```

```
## rsp_correct_solution_cost                                0.30309821
## sppPUsum                                                 0.29245341
## sppPUprod                                                0.25184349
## ig_mean_bottom_bg_redundancy                             0.45583585
## ig_median_bottom_bg_redundancy                           0.45342802
## ig_mean_top_bg_redundancy                                0.43905548
## ig_median_top_bg_redundancy                              0.44245825
## connectance                                              0.32239122
## rsp_realized_FP_rate                                     0.37938952
## rsp_realized_FN_rate                                     0.44236601
## rsp_realized_Ftot_rate                                   0.38767696
## rsp_euc_realized_FP_and_cost_in_err_frac                 0.37938952
## rsp_euc_realized_FN_and_cost_in_err_frac                 0.44236601
## rsp_euc_realized_Ftot_and_cost_in_err_frac               0.38767696
## rsr_COR_euc_out_err_frac                                 1.00000000
## err_mag                                                  0.02913253
##                                                 err_mag
## rsp_num_PUs                                  0.31265801
## rsp_num_spp                                  0.34341715
## rsp_num_spp_per_PU                           0.29516290
## rsp_correct_solution_cost                    0.30472485
## sppPUsum                                     0.33574529
## sppPUprod                                    0.32284925
## ig_mean_bottom_bg_redundancy                -0.52077066
## ig_median_bottom_bg_redundancy              -0.50718361
## ig_mean_top_bg_redundancy                   -0.52276727
## ig_median_top_bg_redundancy                 -0.50581448
## connectance                                 -0.55230410
## rsp_realized_FP_rate                        -0.50376230
## rsp_realized_FN_rate                         0.19825831
## rsp_realized_Ftot_rate                      -0.50395843
## rsp_euc_realized_FP_and_cost_in_err_frac    -0.50376230
## rsp_euc_realized_FN_and_cost_in_err_frac     0.19825831
## rsp_euc_realized_Ftot_and_cost_in_err_frac -0.50395843
## rsr_COR_euc_out_err_frac                     0.02913253
## err_mag                                      1.00000000
```

```r
    #  Plot the correlation matrix.

corrplot (correlations, tl.cex=0.5
#          , order = "hclust"    #  This fails because of NAs or NaNs in the data...
          )
```



8.1.2 Revised set of variables to examine for correlations

```
new_continuous_vars_working_df =
    select (working_df, #c (

            #  Input descriptors
                  rsp_num_PUs,
                  rsp_num_spp,
                  rsp_num_spp_per_PU,
                  rsp_correct_solution_cost,

                  sppPUsum,
                  sppPUprod,

            #  Post-gen knowable problem descriptors

                #  igraph package metrics
                  ig_top,
                  ig_bottom,
                  ig_num_edges_m,
                  ig_ktop,
                  ig_kbottom,
                  ig_bidens,
                  ig_lcctop,
                  ig_lccbottom,
```

```
    ig_distop,
    ig_disbottom,
    ig_cctop,
    ig_ccbottom,
    ig_cclowdottop,
    ig_cclowdotbottom,
    ig_cctopdottop,
    ig_cctopdotbottom,

    ig_mean_bottom_bg_redundancy,
    ig_median_bottom_bg_redundancy,
    ig_mean_top_bg_redundancy,
    ig_median_top_bg_redundancy,

    ig_user_time,
    ig_system_time,
    ig_elapsed_time,
    ig_user_child_time,
    ig_sys_child_time,

#  bipartite package metrics
    connectance,

    web_asymmetry,
    links_per_PUsAndSpp,
    cluster_coefficient,
    weighted_NODF,
    interaction_strength_asymmetry,
    specialisation_asymmetry,
    linkage_density,
    weighted_connectance,
    Shannon_diversity,
    interaction_evenness,
    Alatalo_interaction_evenness,

    mean.number.of.shared.partners.PUs,
    mean.number.of.shared.partners.Spp,
    cluster.coefficient.PUs,
    cluster.coefficient.Spp,
    niche.overlap.PUs,
    niche.overlap.Spp,
    togetherness.PUs,
    togetherness.Spp,
    C.score.PUs,
    C.score.Spp,
    V.ratio.PUs,
    V.ratio.Spp,
    functional.complementarity.PUs,
    functional.complementarity.Spp,
    partner.diversity.PUs,
    partner.diversity.Spp,
    generality.PUs,
    vulnerability.Spp,
```

```
                    bip_user_time,
                    bip_system_time,
                    bip_elapsed_time,
                    bip_user_child_time,
                    bip_sys_child_time,


                #  Possibly knowable realized input error values
                    rsp_realized_FP_rate,
                    rsp_realized_FN_rate,
                    rsp_realized_Ftot_rate,
                    rsp_euc_realized_FP_and_cost_in_err_frac,
                    rsp_euc_realized_FN_and_cost_in_err_frac,
                    rsp_euc_realized_Ftot_and_cost_in_err_frac,

                #  Results and their errors
                    rs_solution_cost,
                    rs_solution_cost_err_frac,
                    abs_rs_solution_cost_err_frac,

                    rs_over_opt_cost_err_frac_of_possible_overcost,
                    rs_under_opt_cost_err_frac_of_possible_undercost,

                    rsr_COR_euc_out_err_frac,

                    rsr_COR_spp_rep_shortfall,
                    rsr_COR_solution_NUM_spp_covered,
                    rsr_COR_solution_FRAC_spp_covered,

                    err_mag,

                #  Reserve selector run times
                    RS_user_time,
                    RS_system_time,
                    RS_elapsed_time,
                    RS_user_child_time,
                    RS_sys_child_time
                      )
#    )
```

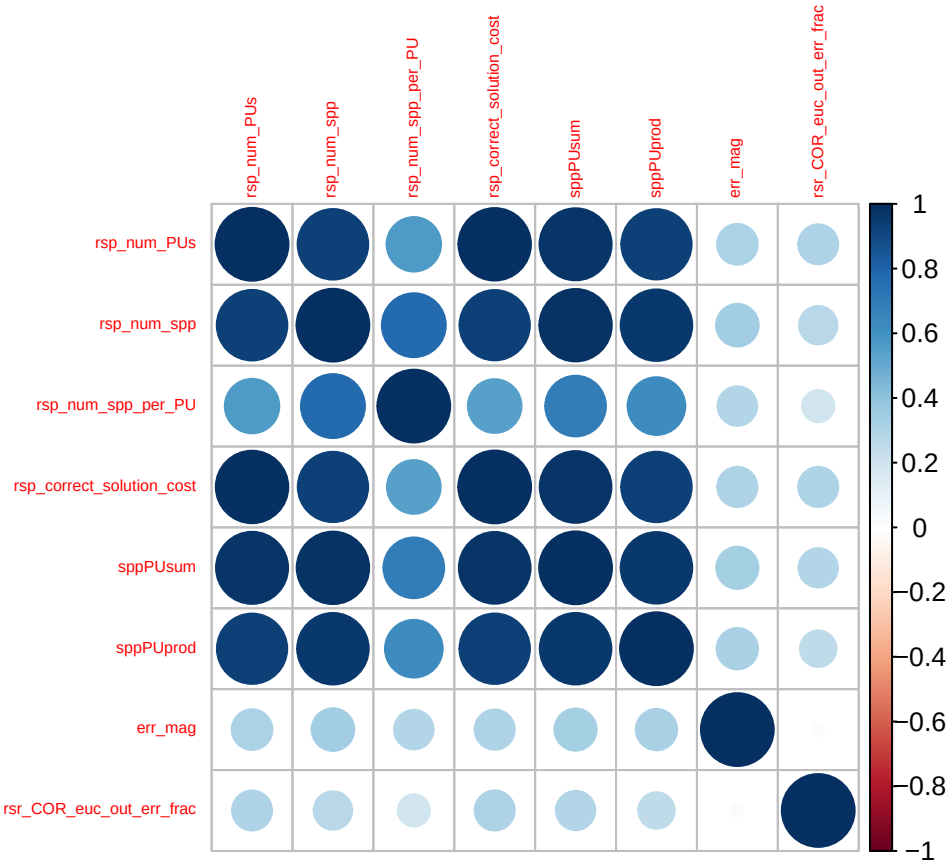
8.1.3 Compute correlations of size vars with output error and magnification

```
size_vars_working_df =
    select (working_df, #c (

            #  Input descriptors
                    rsp_num_PUs,
                    rsp_num_spp,
                    rsp_num_spp_per_PU,
                    rsp_correct_solution_cost,

                    sppPUsum,
                    sppPUprod,
```

```
                #  Results and their errors
                    err_mag,
                    rsr_COR_euc_out_err_frac
                      )

correlations = cor (size_vars_working_df)

dim (correlations)
```

```
## [1] 8 8
```

```
#correlations [, 18:19]

    #  Plot the correlation matrix.

corrplot (correlations, tl.cex=0.5
#           , order = "hclust"    #  This fails because of NAs or NaNs in the data...
          )
```



8.1.4 Compute correlations of igraph metrics with output error and magnification

```
igraph_vars_working_df =
    select (working_df, #c (

                #  igraph package metrics
```

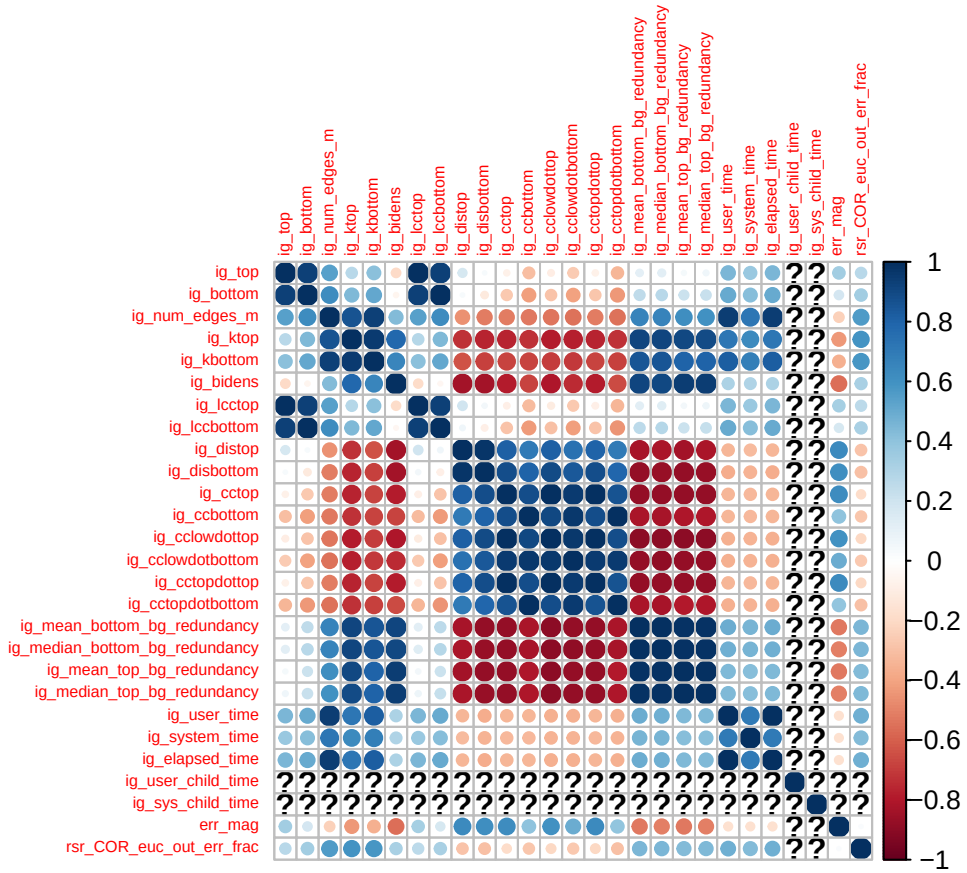
```
                    ig_top,
                    ig_bottom,
                    ig_num_edges_m,
                    ig_ktop,
                    ig_kbottom,
                    ig_bidens,
                    ig_lcctop,
                    ig_lccbottom,
                    ig_distop,
                    ig_disbottom,
                    ig_cctop,
                    ig_ccbottom,
                    ig_cclowdottop,
                    ig_cclowdotbottom,
                    ig_cctopdottop,
                    ig_cctopdotbottom,

                    ig_mean_bottom_bg_redundancy,
                    ig_median_bottom_bg_redundancy,
                    ig_mean_top_bg_redundancy,
                    ig_median_top_bg_redundancy,

                    ig_user_time,
                    ig_system_time,
                    ig_elapsed_time,
                    ig_user_child_time,
                    ig_sys_child_time,

                    err_mag,
                    rsr_COR_euc_out_err_frac,
                      )

correlations = cor (igraph_vars_working_df)
```

```
## Warning in cor(igraph_vars_working_df): the standard deviation is zero
```

```
dim (correlations)
```

```
## [1] 27 27
```

```
#correlations [, 18:19]

    #  Plot the correlation matrix.

corrplot (correlations, tl.cex=0.5
#          , order = "hclust"    #  This fails because of NAs or NaNs in the data...
          )
```



8.1.5 Compute correlations of igraph metrics with output error and magnification

```
bipartite_vars_working_df =
    select (working_df, #c (

                #  bipartite package metrics
                    connectance,

                    web_asymmetry,
                    links_per_PUsAndSpp,
                    cluster_coefficient,
                    weighted_NODF,
                    interaction_strength_asymmetry,
                    specialisation_asymmetry,
                    linkage_density,
                    weighted_connectance,
                    Shannon_diversity,
                    interaction_evenness,
                    Alatalo_interaction_evenness,

                    mean.number.of.shared.partners.PUs,
                    mean.number.of.shared.partners.Spp,
                    cluster.coefficient.PUs,
                    cluster.coefficient.Spp,
```

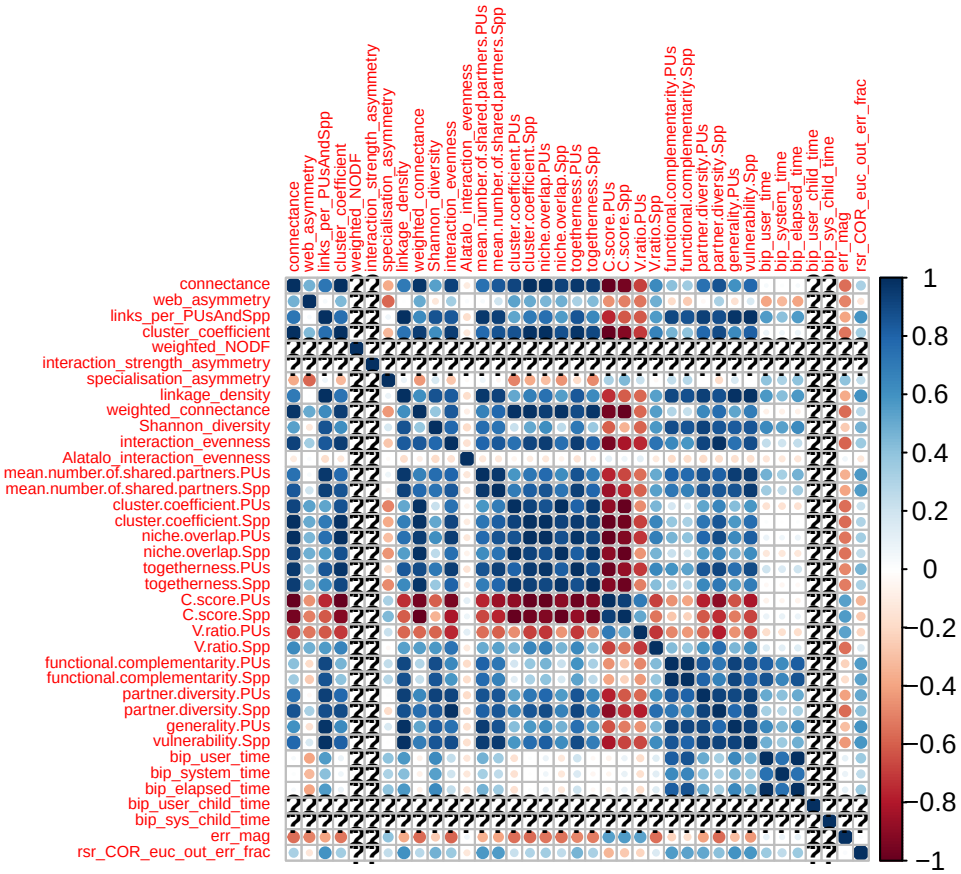
```
                    niche.overlap.PUs,
                    niche.overlap.Spp,
                    togetherness.PUs,
                    togetherness.Spp,
                    C.score.PUs,
                    C.score.Spp,
                    V.ratio.PUs,
                    V.ratio.Spp,
                    functional.complementarity.PUs,
                    functional.complementarity.Spp,
                    partner.diversity.PUs,
                    partner.diversity.Spp,
                    generality.PUs,
                    vulnerability.Spp,

                    bip_user_time,
                    bip_system_time,
                    bip_elapsed_time,
                    bip_user_child_time,
                    bip_sys_child_time,

                #  Results and their errors
                    err_mag,
                    rsr_COR_euc_out_err_frac
                      )

correlations = cor (bipartite_vars_working_df)
```

```
## Warning in cor(bipartite_vars_working_df): the standard deviation is zero
```

```
dim (correlations)
```

```
## [1] 37 37
```

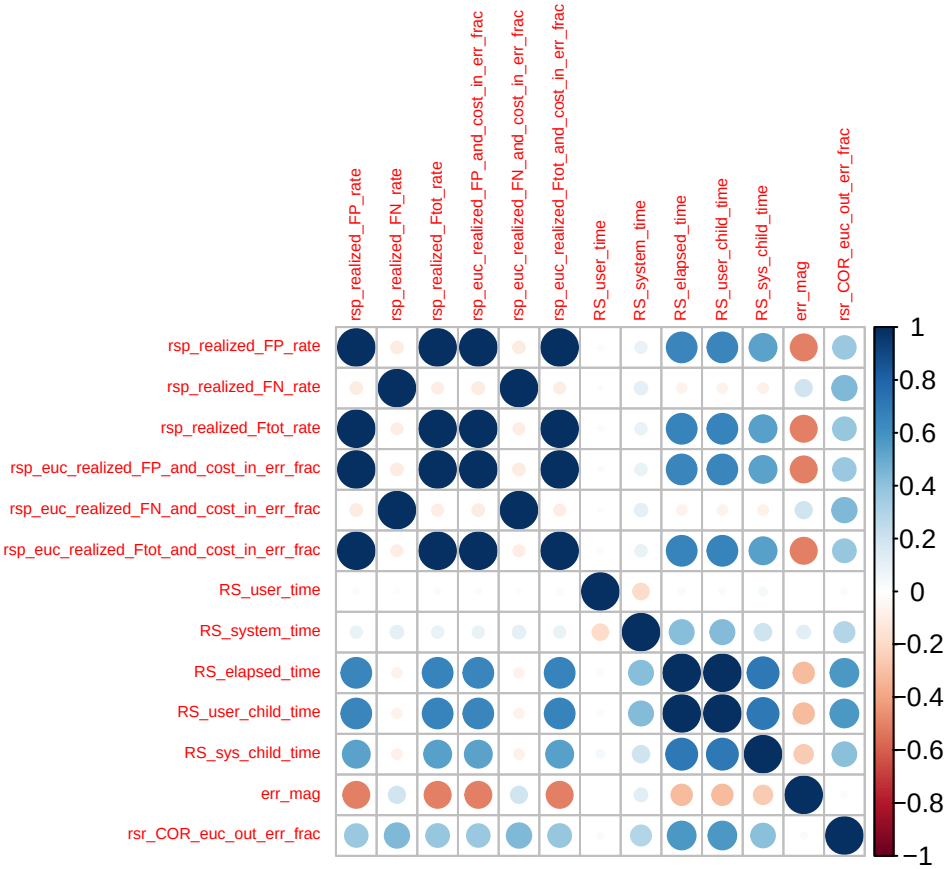
```
#correlations [, 18:19]

    #  Plot the correlation matrix.

corrplot (correlations, tl.cex=0.5
#          , order = "hclust"    #  This fails because of NAs or NaNs in the data...
          )
```




```
possibly_knowable_vars_working_df =
    select (working_df, #c (

                #  Possibly knowable realized input error values
                    rsp_realized_FP_rate,
                    rsp_realized_FN_rate,
                    rsp_realized_Ftot_rate,
                    rsp_euc_realized_FP_and_cost_in_err_frac,
                    rsp_euc_realized_FN_and_cost_in_err_frac,
                    rsp_euc_realized_Ftot_and_cost_in_err_frac,

                #  Reserve selector run times
                    RS_user_time,
                    RS_system_time,
                    RS_elapsed_time,
                    RS_user_child_time,
                    RS_sys_child_time,

                #  Results and their errors
                    err_mag,
                    rsr_COR_euc_out_err_frac
                      )

correlations = cor (possibly_knowable_vars_working_df)

dim (correlations)
```

```
## [1] 13 13
```

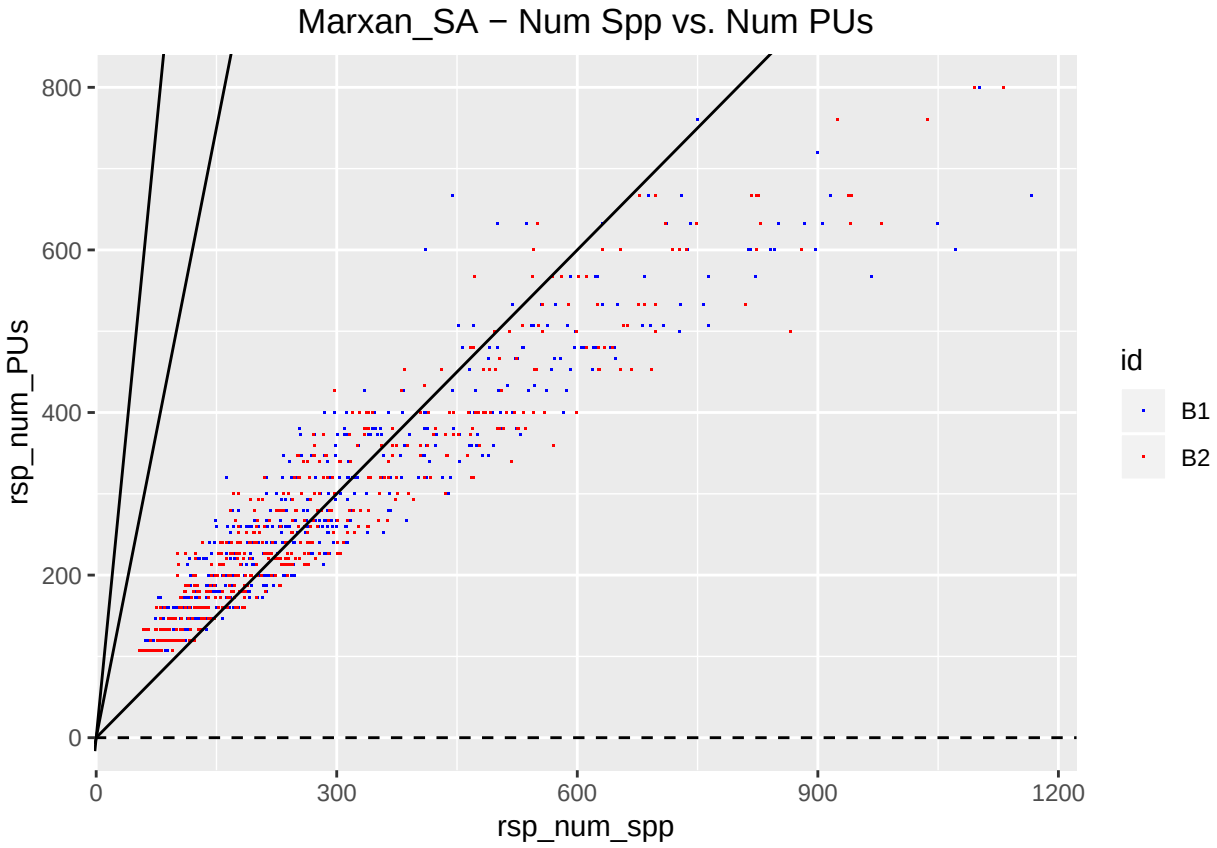
```
#correlations [, 18:19]

    #  Plot the correlation matrix.

corrplot (correlations, tl.cex=0.5
#          , order = "hclust"    #  This fails because of NAs or NaNs in the data...
          )
```



8.2 NOT cheating: Fit prediction using problem size

8.2.1 Plot num PUs vs. num spp to show range of problem sizes

```
gg_numSpp_vs_numPUs_all_on_1 (rs_name,
                                                  #####easyHard_df,
                              working_df,
                                                  ref_y = 0

                                                  # , shape_type="."
                                                  # , alpha_level=1
                                                  )
```



8.2.2 Compute the fit of prediction to problem size

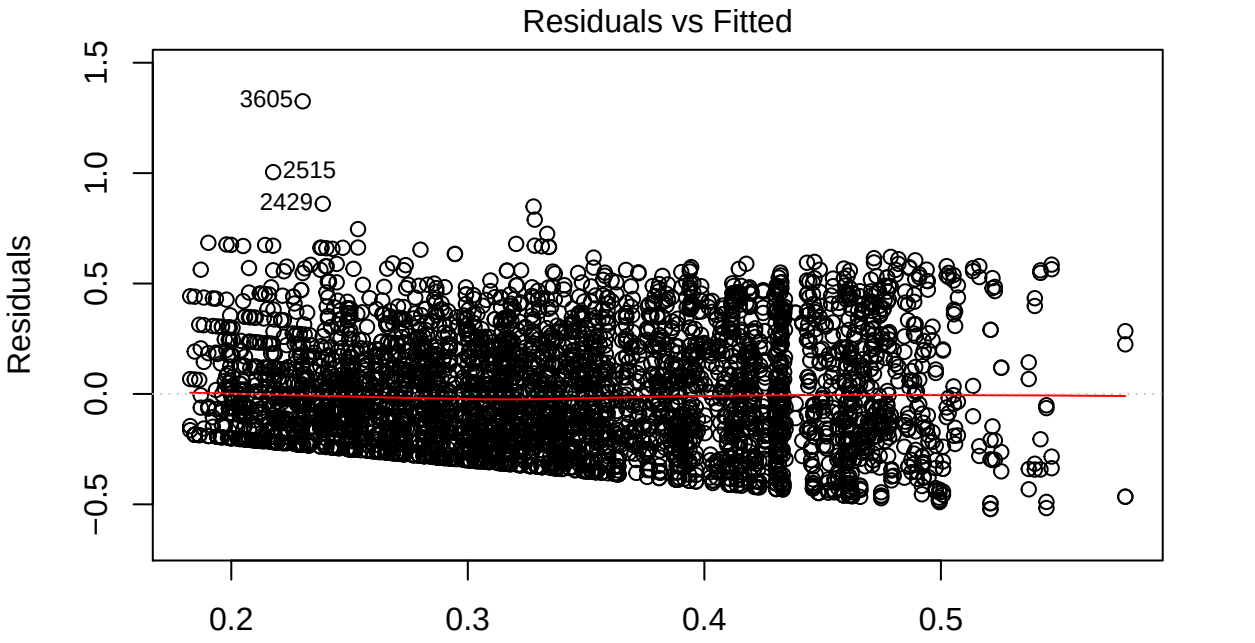
```
cat ("\n...  NOT cheating:  Fit prediction using problem size  ...\n")

            # #  Input descriptors
            #         rsp_num_PUs,
            #         rsp_num_spp,
            #         rsp_num_spp_per_PU,
            #         rsp_correct_solution_cost,
            #
            #         sppPUsum,
            #         sppPUprod,

easyHard_lm_fit = lm (rsr_COR_euc_out_err_frac ~
                          rsp_num_PUs + rsp_num_spp +
                          rsp_num_spp_per_PU + sppPUprod,
                      easyHard_df)

summary (easyHard_lm_fit)
plot (easyHard_lm_fit)

# easyHard_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df

easyHard_not_huge_out_err_lm_fit =
    lm (rsr_COR_euc_out_err_frac ~ rsp_num_PUs + rsp_num_spp +
```

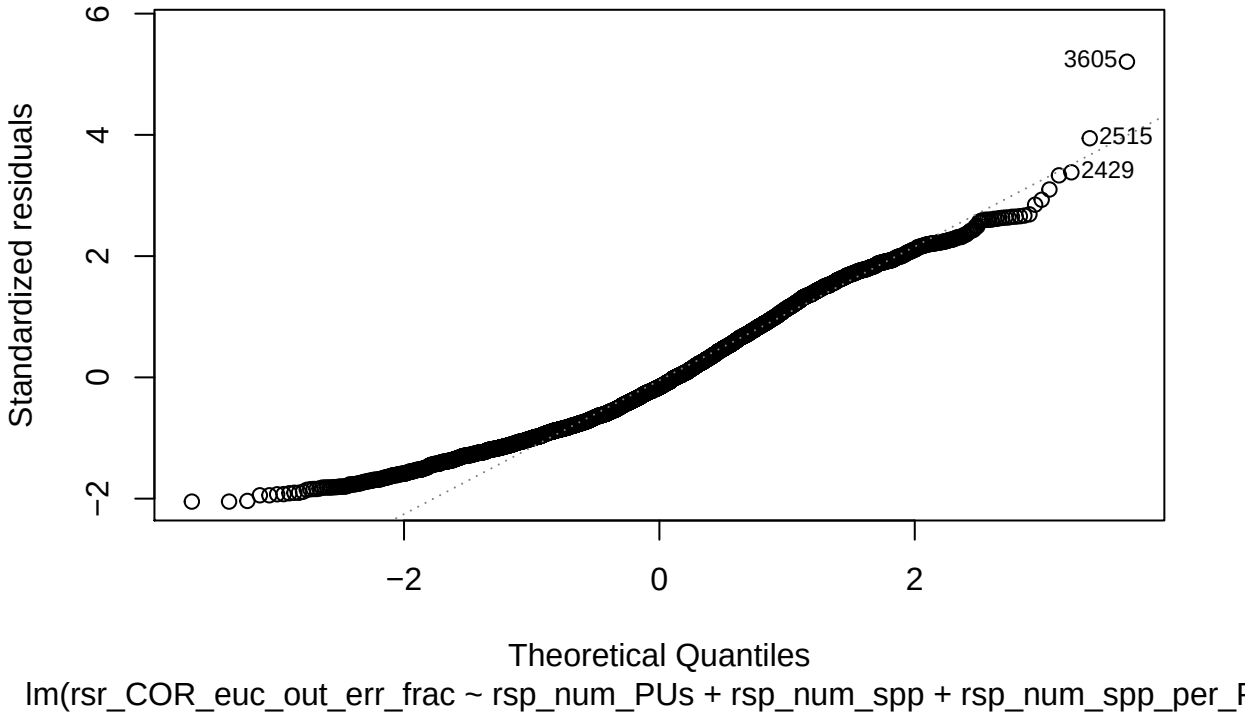
```
                                    rsp_num_spp_per_PU + sppPUprod,
#####        easyHard_not_huge_out_err_df)
        working_df)

summary (easyHard_not_huge_out_err_lm_fit)
plot (easyHard_not_huge_out_err_lm_fit)


#####pred_easyHard_not_huge_out_err_lm_fit =
#####    predict (easyHard_not_huge_out_err_lm_fit,
#####             easyHard_not_huge_out_err_df)

#####plot (easyHard_not_huge_out_err_df$rsr_COR_euc_out_err_frac,
#####      pred_easyHard_not_huge_out_err_lm_fit,
#####      xlab="rsr_COR_euc_out_err_frac", ylab="Predictions", main="COR OutErr vs. Fit Values from Pr
```

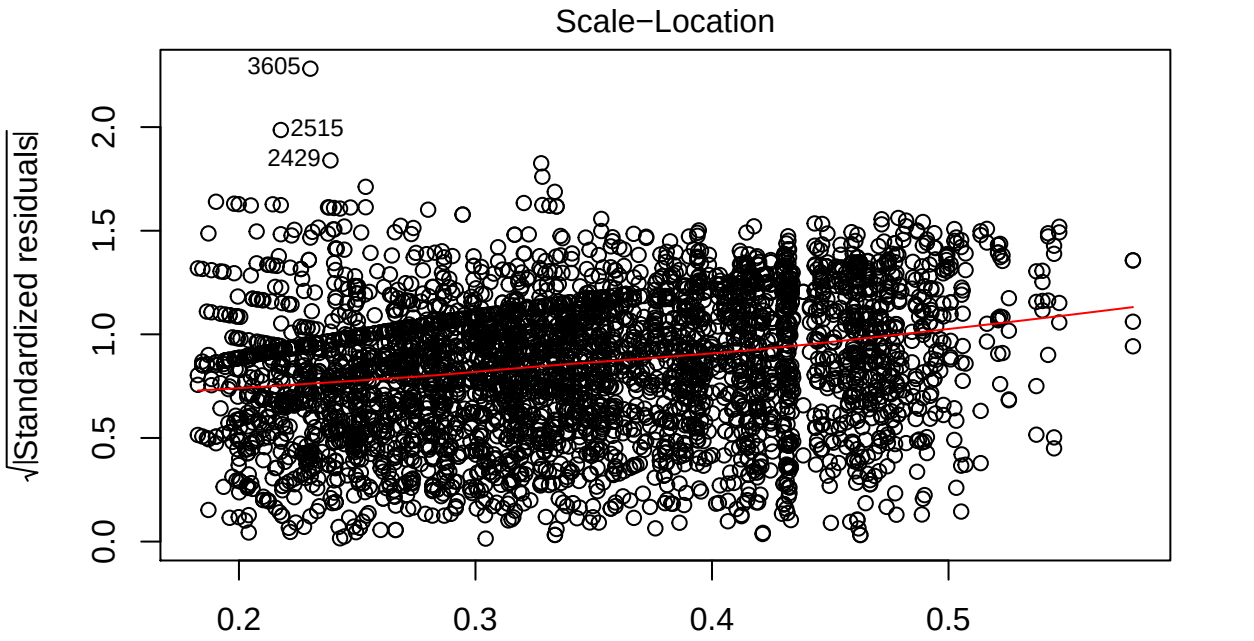
```
working_lm_fit = lm (rsr_COR_euc_out_err_frac ~
                         rsp_num_PUs + rsp_num_spp +
                         rsp_num_spp_per_PU + sppPUprod,
                     working_df)

summary (working_lm_fit)
```

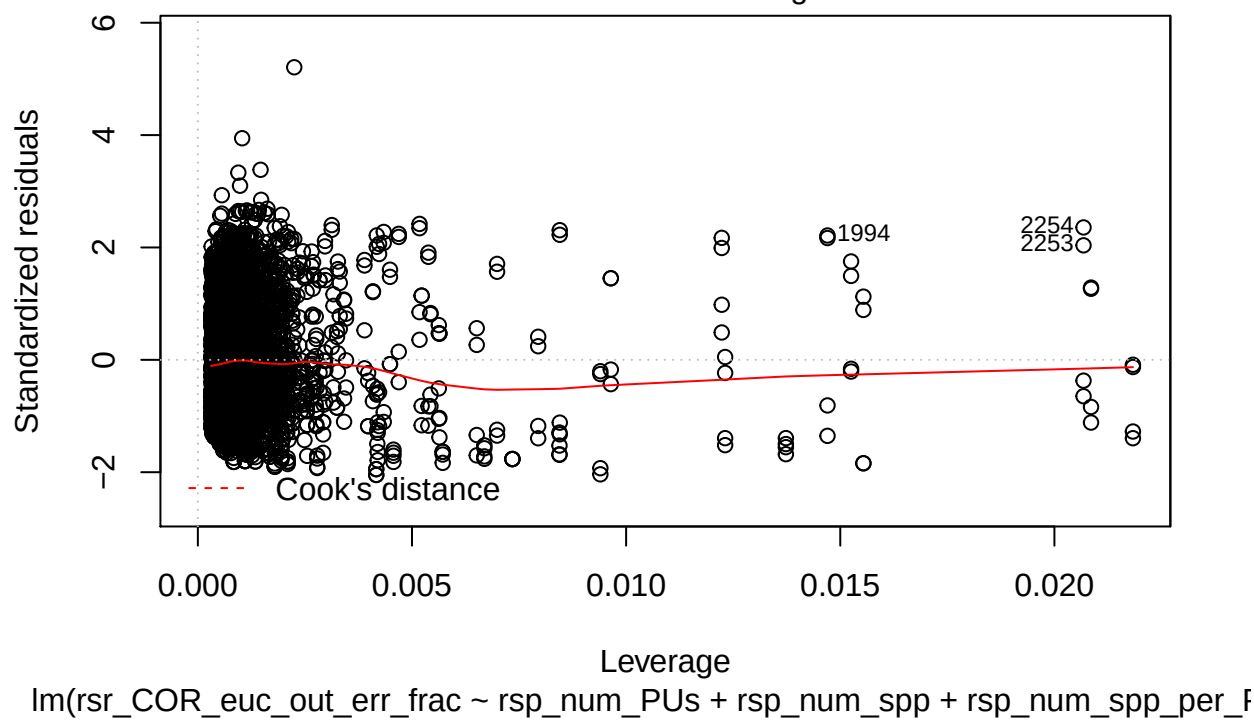
```
##
## Call:
## lm(formula = rsr_COR_euc_out_err_frac ~ rsp_num_PUs + rsp_num_spp +
##     rsp_num_spp_per_PU + sppPUprod, data = working_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52094 -0.20185 -0.04104  0.17697  1.32537
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         4.682e-02  4.458e-02   1.050 0.293706
## rsp_num_PUs         9.462e-04  1.492e-04   6.340 2.56e-10 ***
## rsp_num_spp         1.236e-04  2.432e-04   0.508 0.611306
## rsp_num_spp_per_PU  6.285e-02  6.099e-02   1.031 0.302768
## sppPUprod          -6.641e-07  1.854e-07  -3.581 0.000346 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2548 on 3991 degrees of freedom
## Multiple R-squared:  0.1018, Adjusted R-squared:  0.1009
## F-statistic: 113.1 on 4 and 3991 DF,  p-value: < 2.2e-16
```

```
plot (working_lm_fit)
```


Residuals vs Fitted
Residuals
Fitted values
lm(rsr_COR_euc_out_err_frac ~ rsp_num_PUs + rsp_num_spp + rsp_num_spp_per_F
3605
2515
2429
1.5
1.0
0.5
0.0
−0.5
0.2
0.3
0.4
0.5


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(rsr_COR_euc_out_err_frac ~ rsp_num_PUs + rsp_num_spp + rsp_num_spp_per_F
3605
2515
2429
6
4
2
0
−2
−2
0
2






```
# working_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df

#####working_not_huge_out_err_lm_fit =
#####    lm (rsr_COR_euc_out_err_frac ~ rsp_num_PUs + rsp_num_spp +
#####                                   rsp_num_spp_per_PU + sppPUprod,
```

```
#####        working_not_huge_out_err_df)

#####summary (working_not_huge_out_err_lm_fit)
#####plot (working_not_huge_out_err_lm_fit)
```

8.3 Graph measures help

8.3.1 Without cheating

- Graph measures do broadly predict trend in output error, but still lots of variability.
- What about with problem size information too?

```
cat ("\n...  NOT cheating:  Fit prediction using graph measures, problem size  ...\n")
```

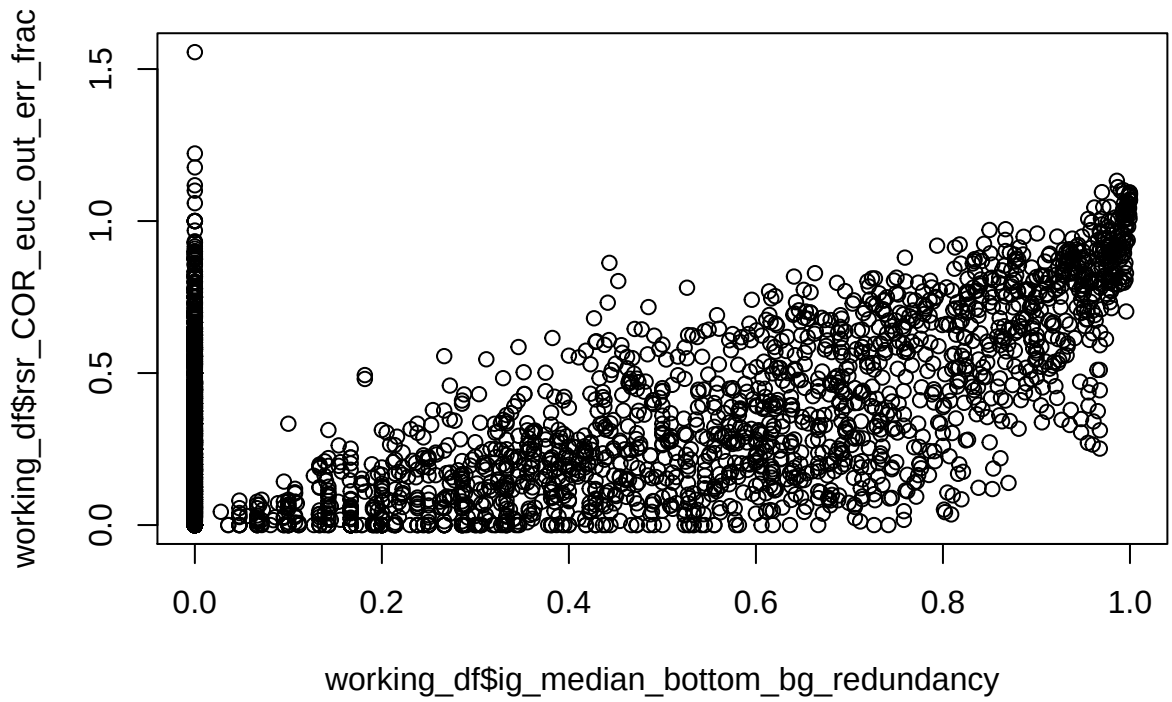
```
##
## ...  NOT cheating:  Fit prediction using graph measures, problem size  ...
```

```
                #  igraph package metrics
            #                ig_rsp_UUID,

                    # ig_top,
                    # ig_bottom,
                    # ig_num_edges_m,
                    # ig_ktop,
                    # ig_kbottom,
                    # ig_bidens,
                    # ig_lcctop,
                    # ig_lccbottom,
                    # ig_distop,
                    # ig_disbottom,
                    # ig_cctop,
                    # ig_ccbottom,
                    # ig_cclowdottop,
                    # ig_cclowdotbottom,
                    # ig_cctopdottop,
                    # ig_cctopdotbottom,
                    # ig_mean_bottom_bg_redundancy,
                    # ig_median_bottom_bg_redundancy,
                    # ig_mean_top_bg_redundancy,
                    # ig_median_top_bg_redundancy,

                            # ig_user_time,
                            # ig_system_time,
                            # ig_elapsed_time,
                            # ig_user_child_time,
                            # ig_sys_child_time,

                #  bipartite package metrics
            #                 bip_rsp_UUID,

                    # connectance,
```

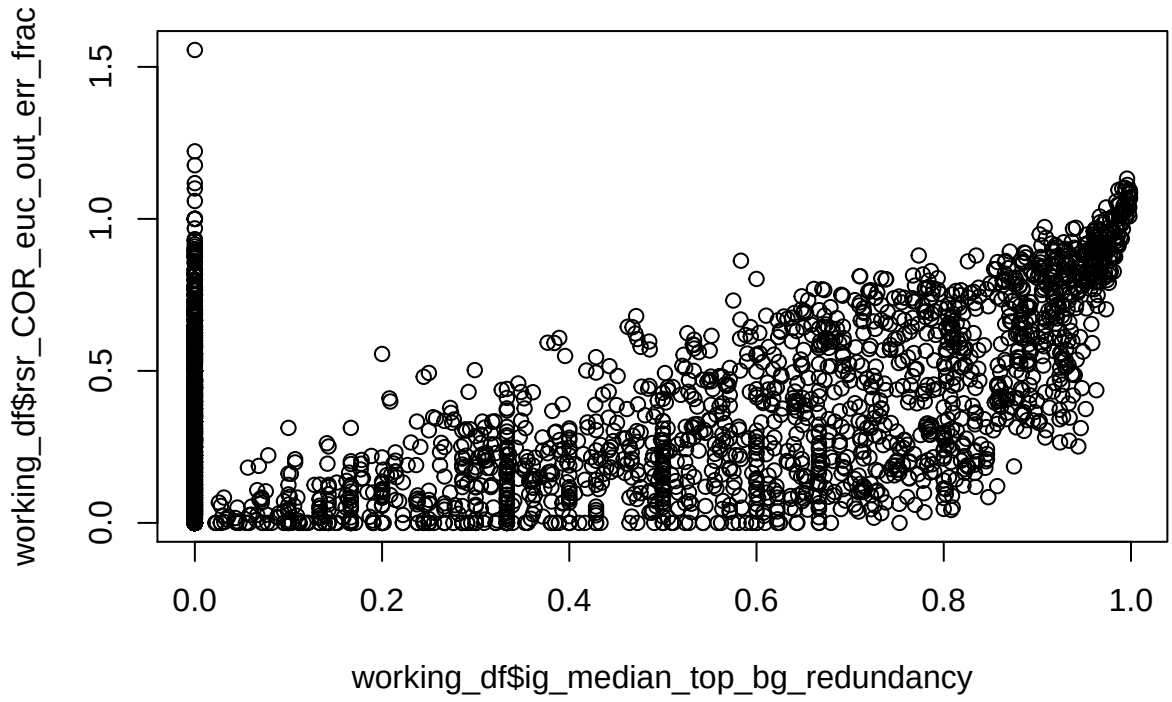
8.3.2 Quick and dirty plot of redundancy vs. output error fraction

Still lots of variance at every level, however, redundancy does seem to put a lower bound on the amount of input error in this data, i.e., you can bet that your output error will be **at least** this much. That seems pretty useful even if you can't predict the exact error amount.

```
plot (working_df$ig_median_bottom_bg_redundancy, working_df$rsr_COR_euc_out_err_frac)
```




```
plot (working_df$ig_median_top_bg_redundancy, working_df$rsr_COR_euc_out_err_frac)
```



```
                # ig_mean_bottom_bg_redundancy,
                # ig_median_bottom_bg_redundancy,
                # ig_mean_top_bg_redundancy,
                # ig_median_top_bg_redundancy,

                        # ig_user_time,
                        # ig_system_time,
                        # ig_elapsed_time,
                        # ig_user_child_time,
                        # ig_sys_child_time,

            #  bipartite package metrics
        #                bip_rsp_UUID,

                # connectance,
```

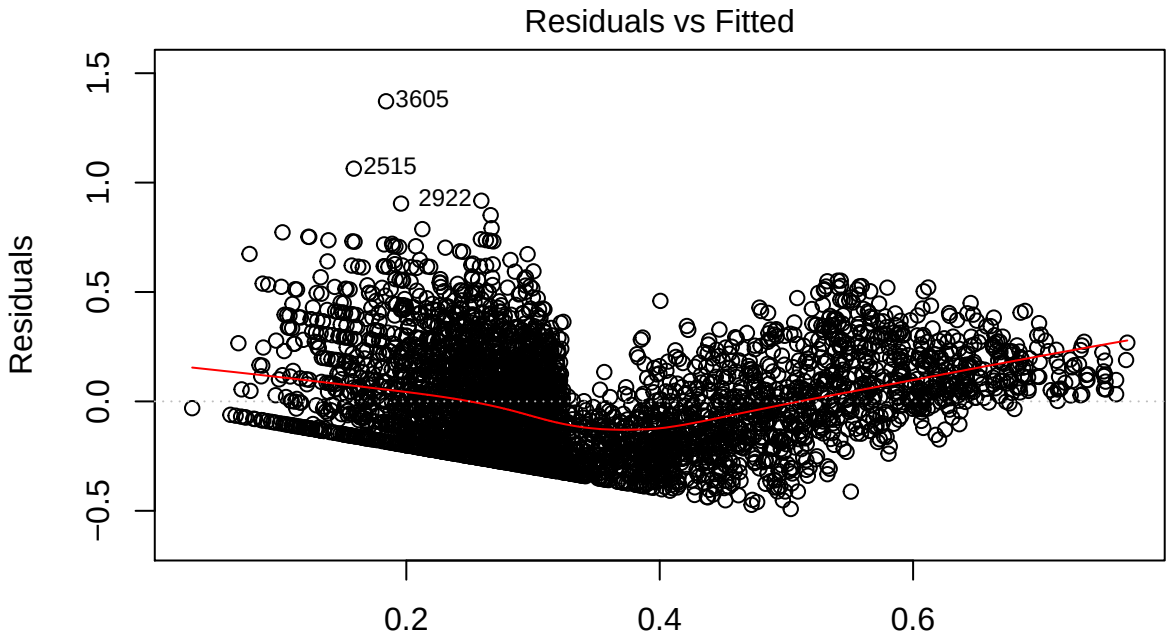
8.3.3 Compute the fit of prediction to graph measures (redundancy, connectance)

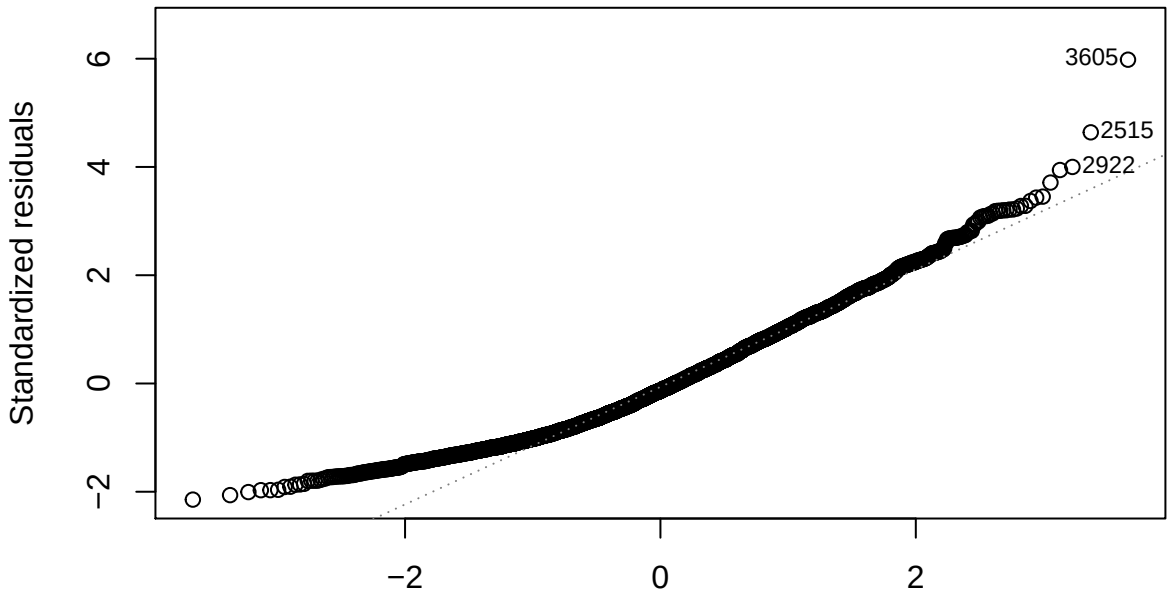
8.3.4 Plot Working COR OutErr vs. Fit Values from Graph Measures

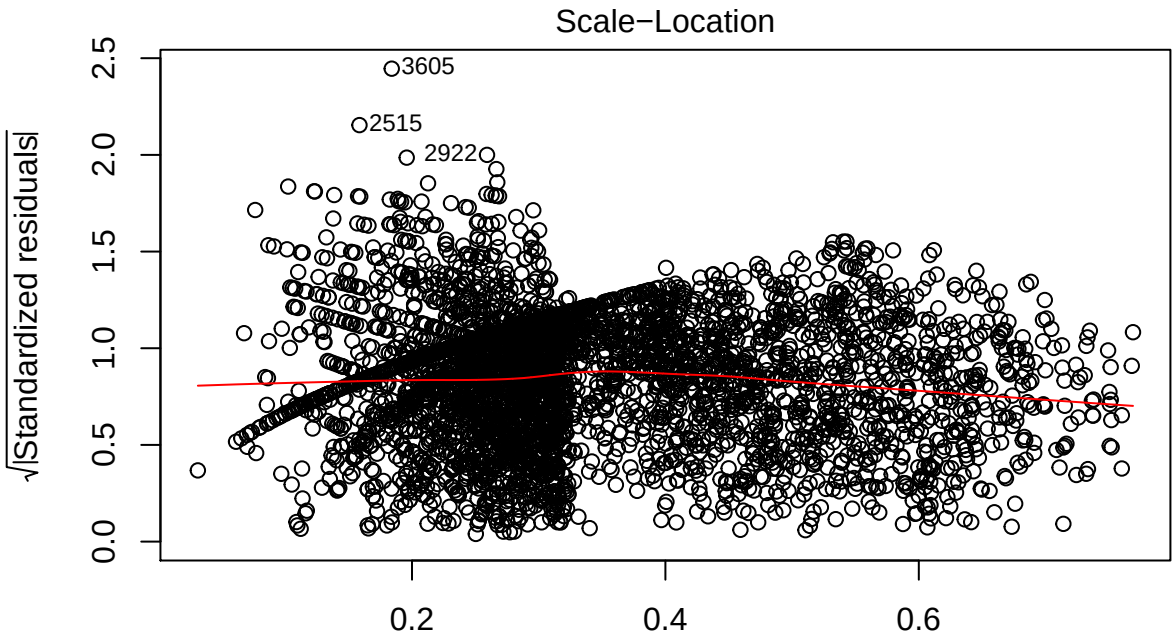
```
working_lm_fit = lm (rsr_COR_euc_out_err_frac ~
                        ig_median_bottom_bg_redundancy + ig_median_top_bg_redundancy +
                        connectance,
                    working_df)

summary (working_lm_fit)
```

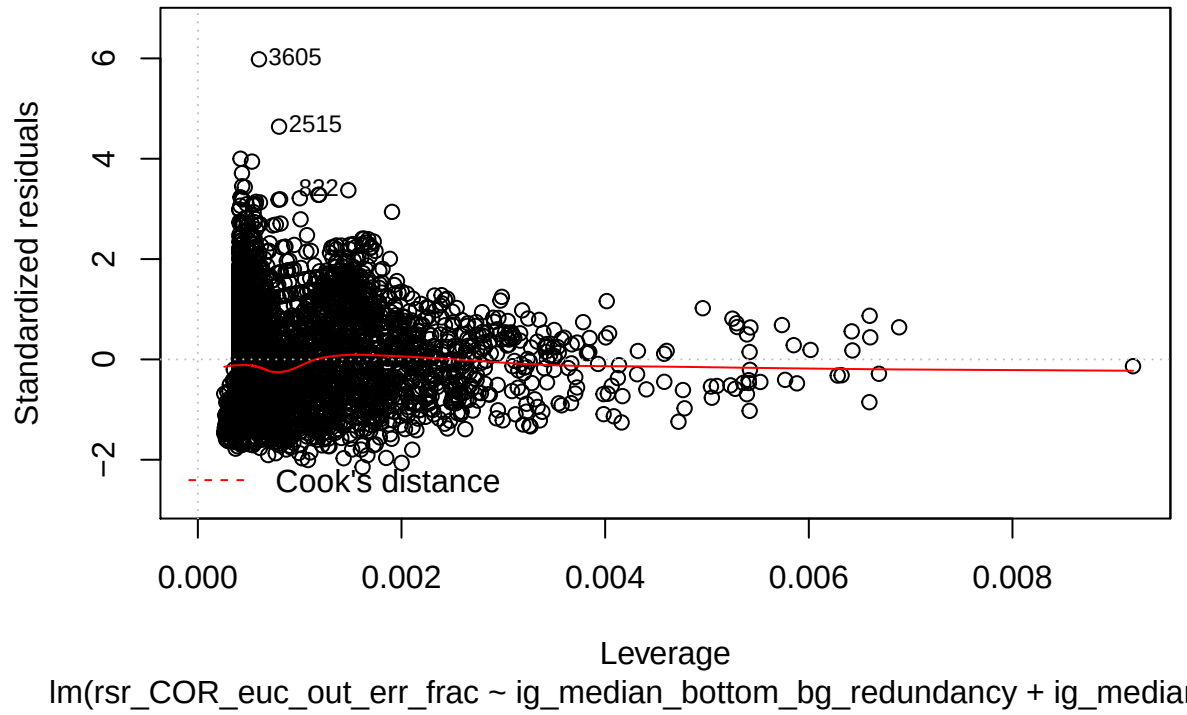
```
##
## Call:
## lm(formula = rsr_COR_euc_out_err_frac ~ ig_median_bottom_bg_redundancy +
##     ig_median_top_bg_redundancy + connectance, data = working_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49158 -0.18213 -0.03088  0.15335  1.37157
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.353979   0.007416  47.731   <2e-16 ***
## ig_median_bottom_bg_redundancy 0.021143   0.073088   0.289    0.772
## ig_median_top_bg_redundancy    0.888330   0.086298  10.294   <2e-16 ***
## connectance                   -7.012475   0.370527 -18.926   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2294 on 3992 degrees of freedom
## Multiple R-squared:  0.2721, Adjusted R-squared:  0.2716
## F-statistic: 497.5 on 3 and 3992 DF,  p-value: < 2.2e-16
```

```
plot (working_lm_fit)
```


Residuals vs Fitted
Residuals
Fitted values
lm(rsr_COR_euc_out_err_frac ~ ig_median_bottom_bg_redundancy + ig_median_to
3605
2515
2922
1.5
1.0
0.5
0.0
−0.5
0.2
0.4
0.6

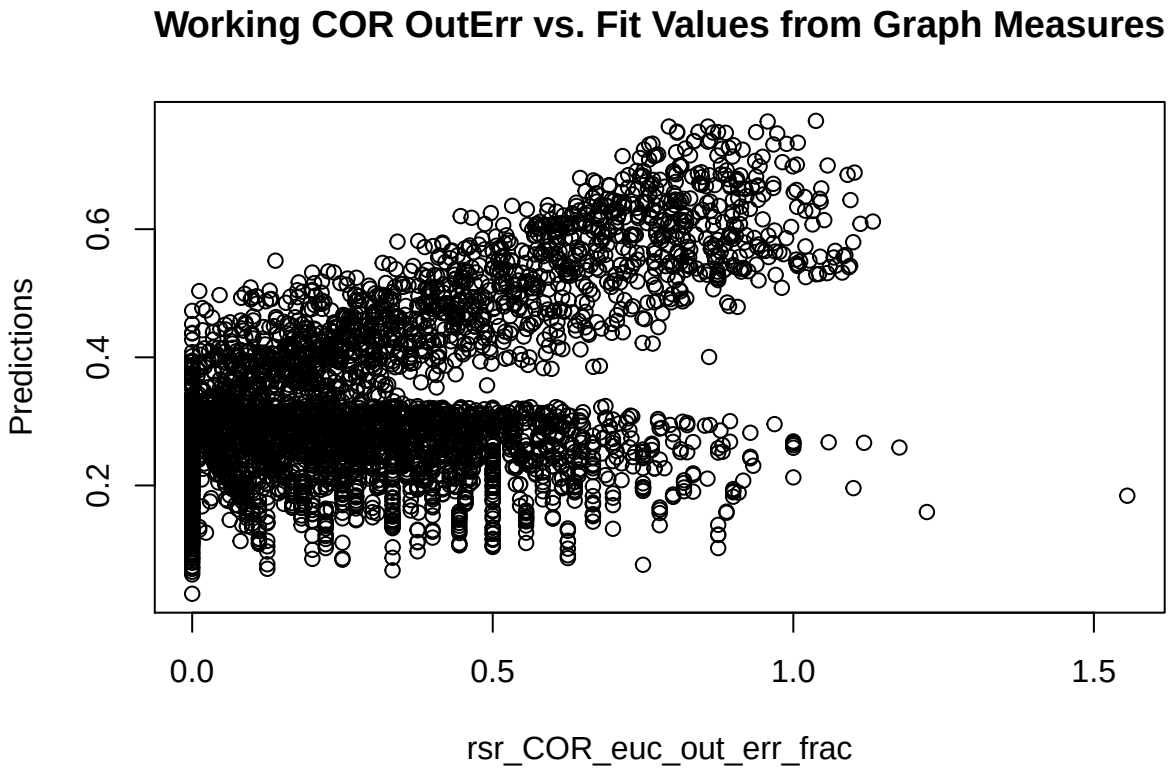

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(rsr_COR_euc_out_err_frac ~ ig_median_bottom_bg_redundancy + ig_median_to
3605
2515
2922
6
4
2
0
−2
−2
0
2






```
pred_working_lm_fit =
    predict (working_lm_fit,
             working_df)

plot (working_df$rsr_COR_euc_out_err_frac,
      pred_working_lm_fit,
```

```
      xlab="rsr_COR_euc_out_err_frac", ylab="Predictions", main="Working COR OutErr vs. Fit Values from
```



8.3.4.1 Again, without cheating, graph measures + problem size data

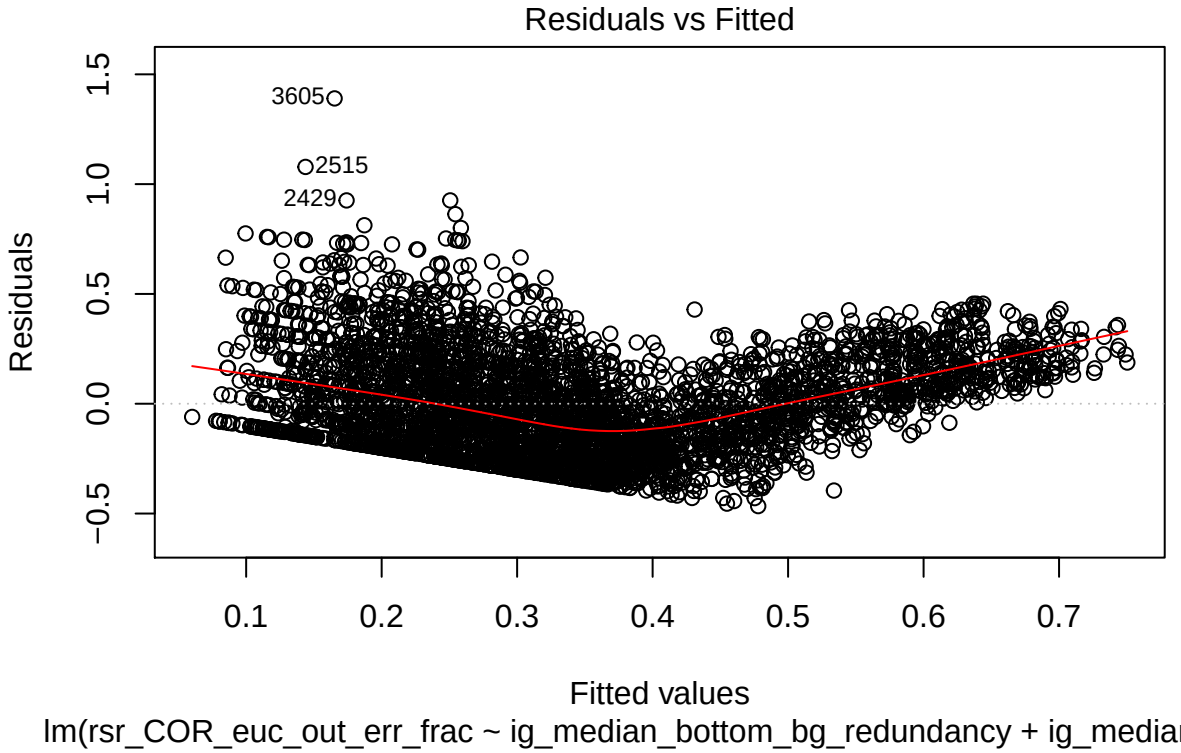
```
easyHard_lm_fit = lm (rsr_COR_euc_out_err_frac ~
                          ig_median_bottom_bg_redundancy + ig_median_top_bg_redundancy +
                          connectance +
                              rsp_num_PUs + rsp_num_spp +
                                   rsp_num_spp_per_PU + sppPUprod,
                      easyHard_df)

summary (easyHard_lm_fit)
plot (easyHard_lm_fit)
```

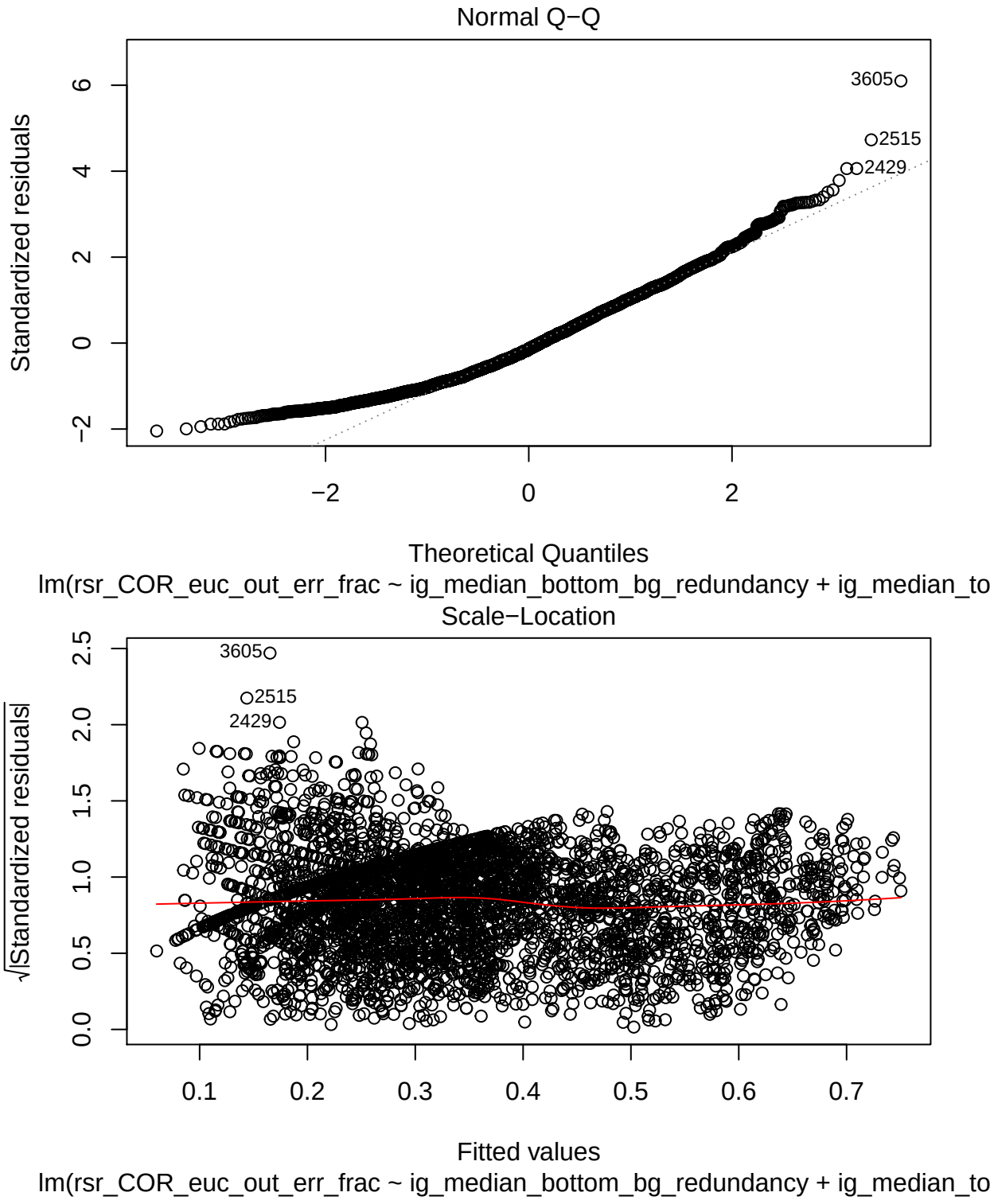
```
working_lm_fit = lm (rsr_COR_euc_out_err_frac ~
                          ig_median_bottom_bg_redundancy + ig_median_top_bg_redundancy +
                          connectance +
                              rsp_num_PUs + rsp_num_spp +
                                   rsp_num_spp_per_PU + sppPUprod,
                      working_df)

summary (working_lm_fit)
```

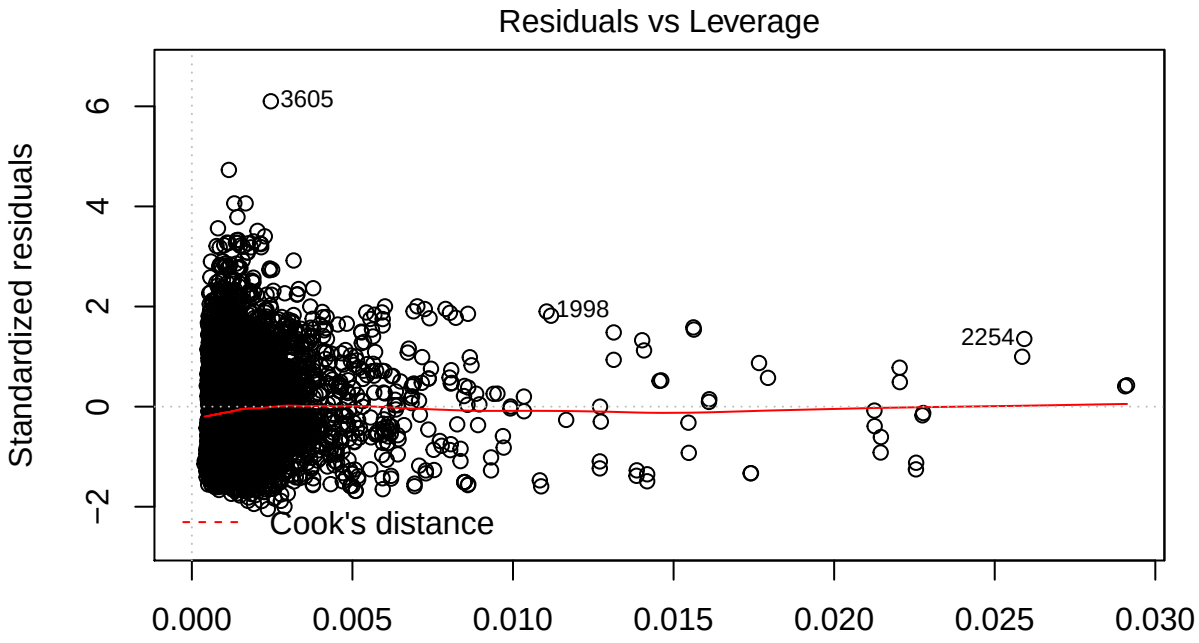
```
##
## Call:
## lm(formula = rsr_COR_euc_out_err_frac ~ ig_median_bottom_bg_redundancy +
##     ig_median_top_bg_redundancy + connectance + rsp_num_PUs +
##     rsp_num_spp + rsp_num_spp_per_PU + sppPUprod, data = working_df)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46610 -0.18244 -0.03108  0.15393  1.39029
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.277e-01  5.206e-02   2.454 0.014190 *
## ig_median_bottom_bg_redundancy -4.280e-02  8.379e-02  -0.511 0.609564
## ig_median_top_bg_redundancy     6.324e-01  1.023e-01   6.184 6.88e-10 ***
## connectance                    -3.278e+00  6.760e-01  -4.850 1.28e-06 ***
## rsp_num_PUs                     5.301e-04  1.482e-04   3.576 0.000353 ***
## rsp_num_spp                    -2.103e-05  2.197e-04  -0.096 0.923744
## rsp_num_spp_per_PU              5.838e-02  5.704e-02   1.023 0.306158
## sppPUprod                      -2.610e-07  1.761e-07  -1.482 0.138414
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2282 on 3988 degrees of freedom
## Multiple R-squared:  0.2803, Adjusted R-squared:  0.2791
## F-statistic: 221.9 on 7 and 3988 DF,  p-value: < 2.2e-16
```

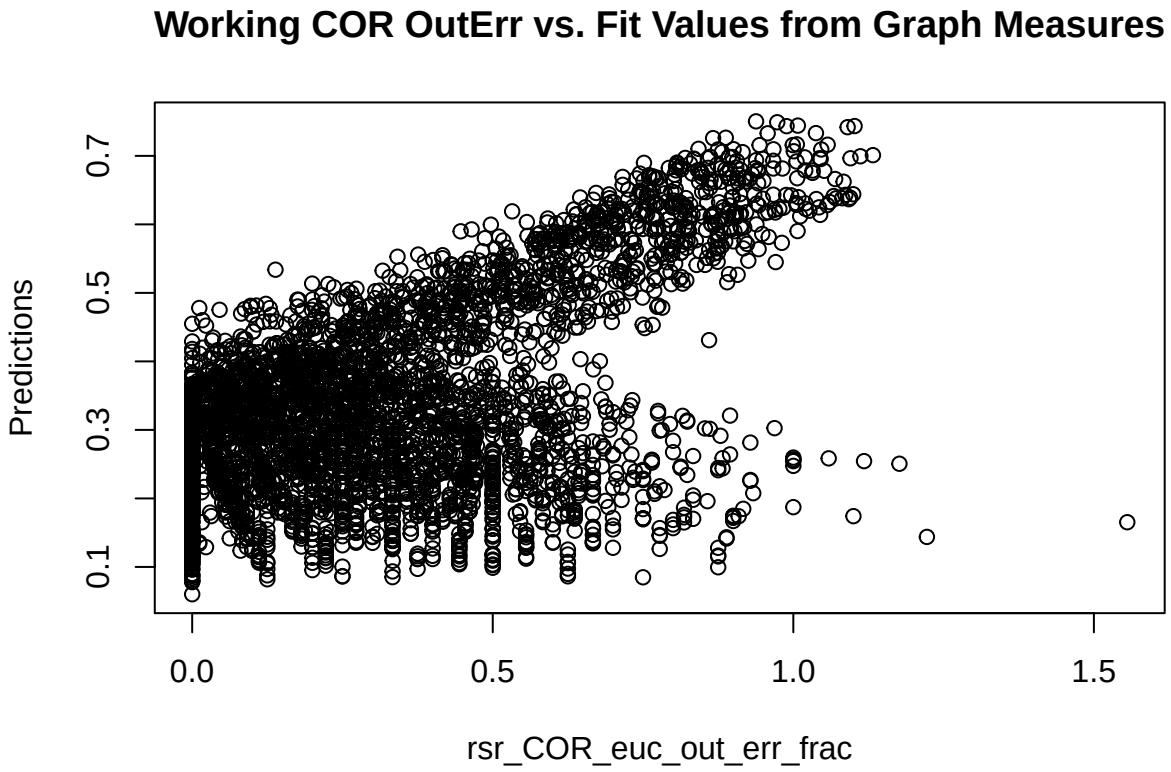
```
plot (working_lm_fit)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(rsr_COR_euc_out_err_frac ~ ig_median_bottom_bg_redundancy + ig_median_to
3605
2515
2429
Scale−Location
√|Standardized residuals|
Fitted values
lm(rsr_COR_euc_out_err_frac ~ ig_median_bottom_bg_redundancy + ig_median_to




```
pred_working_lm_fit =
    predict (working_lm_fit,
             working_df)

plot (working_df$rsr_COR_euc_out_err_frac,
      pred_working_lm_fit,
      xlab="rsr_COR_euc_out_err_frac", ylab="Predictions", main="Working COR OutErr vs. Fit Values from
```



8.3.5 With cheating

- Generally, output error does increase with input error, but still has large variability.
- Did combination of the two predict better?

8.3.5.1 Cheating, using input errors only

```
cat ("\n...  CHEATING:  Fit prediction using input error  ...\n")
```

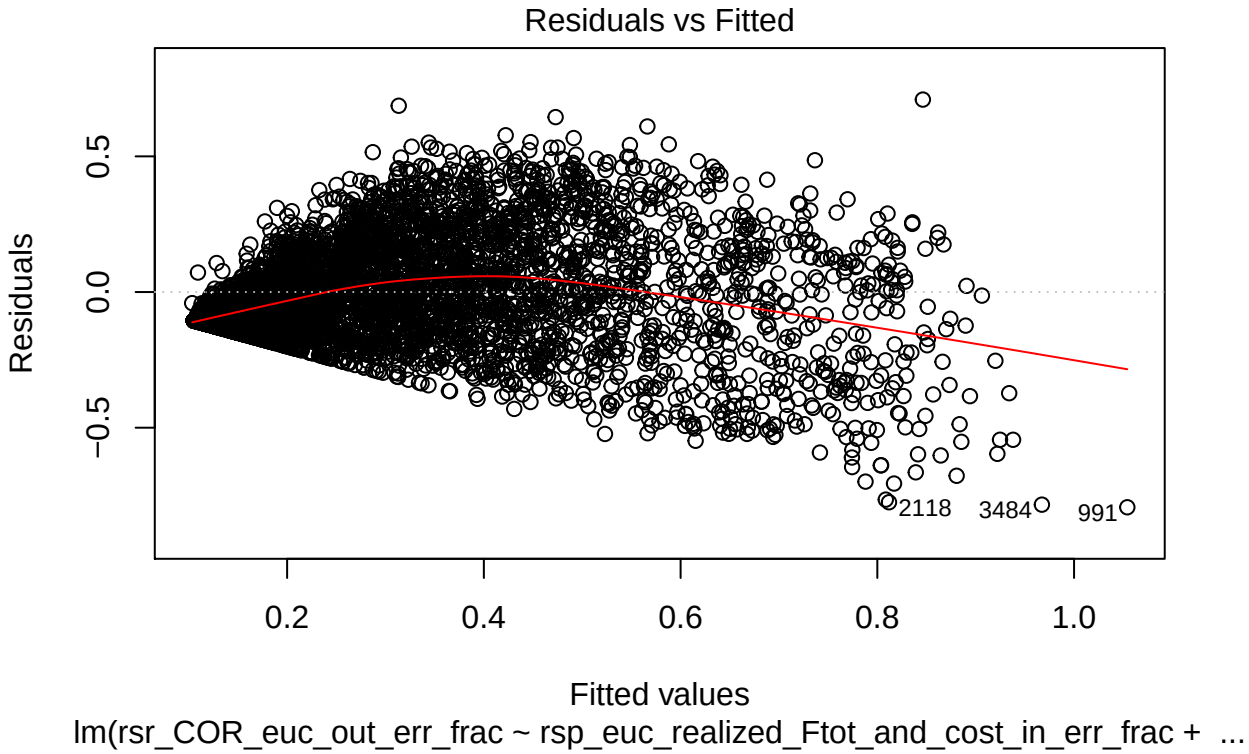
```
##
## ...  CHEATING:  Fit prediction using input error  ...
```

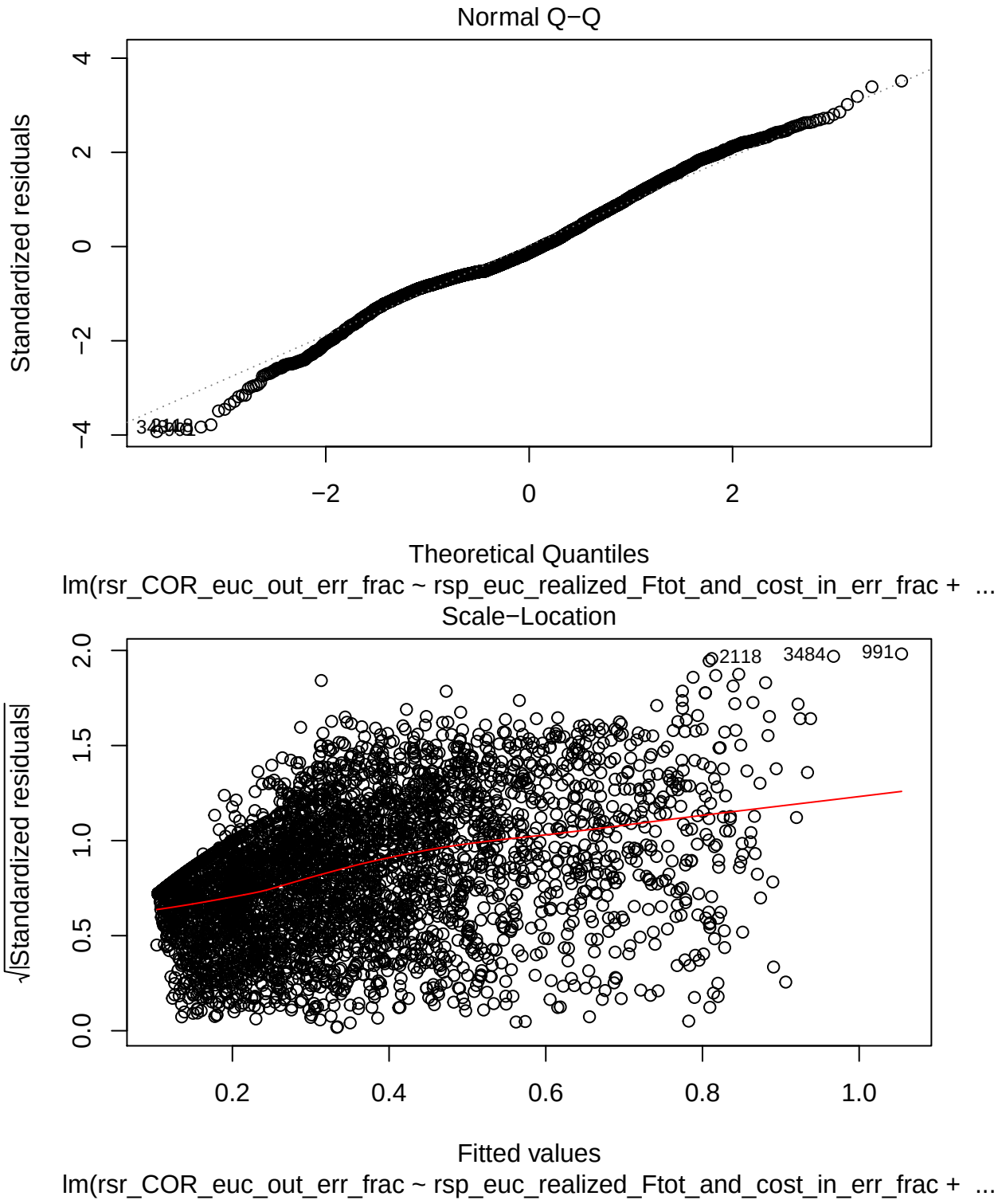
```
working_lm_fit = lm (rsr_COR_euc_out_err_frac ~ rsp_euc_realized_Ftot_and_cost_in_err_frac +
                                    rsp_realized_FP_rate +
                                    rsp_realized_FN_rate +
                                    rsp_realized_Ftot_rate +
                                    rsp_euc_realized_FP_and_cost_in_err_frac +
                                    rsp_euc_realized_FN_and_cost_in_err_frac,
                     working_df)

summary (working_lm_fit)
```

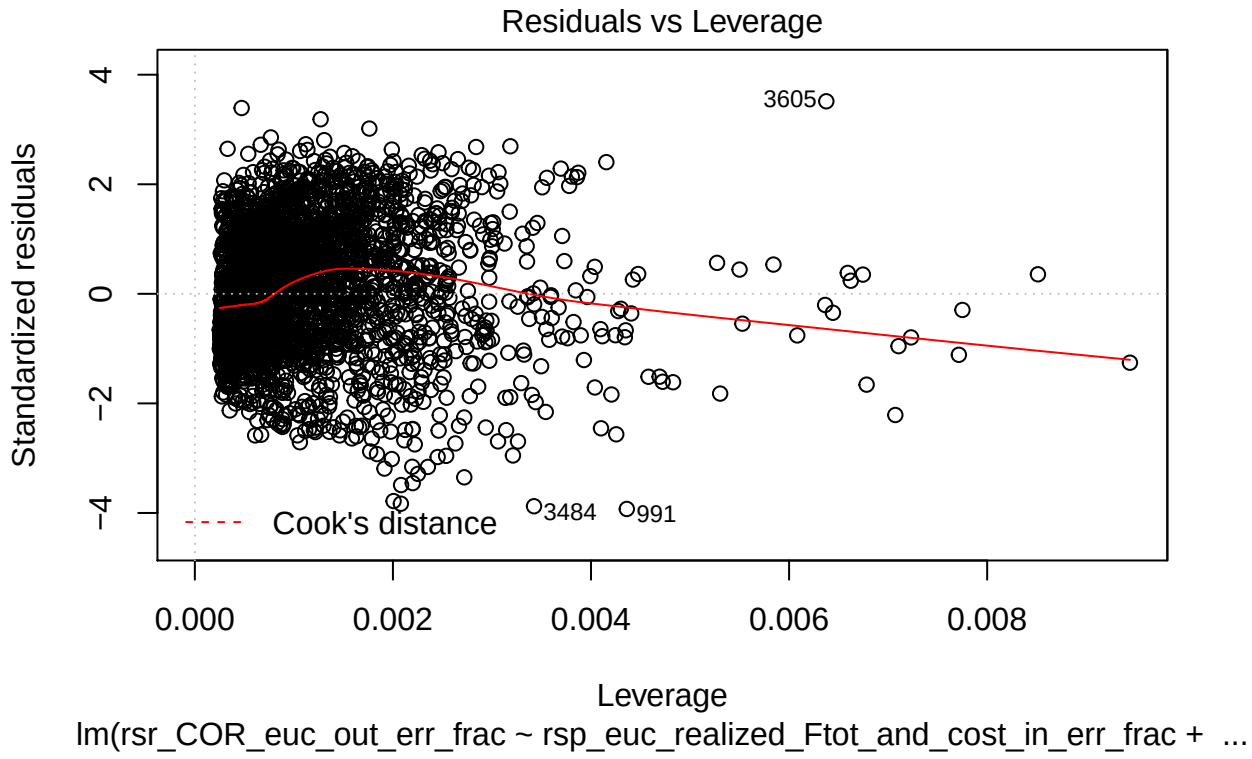
```
##
## Call:
## lm(formula = rsr_COR_euc_out_err_frac ~ rsp_euc_realized_Ftot_and_cost_in_err_frac +
##     rsp_realized_FP_rate + rsp_realized_FN_rate + rsp_realized_Ftot_rate +
##     rsp_euc_realized_FP_and_cost_in_err_frac + rsp_euc_realized_FN_and_cost_in_err_frac,
##     data = working_df)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79327 -0.12593 -0.02584  0.13288  0.70926
##
## Coefficients: (3 not defined because of singularities)
##                                             Estimate Std. Error t value
## (Intercept)                                1.052e-01  5.619e-03  18.719
## rsp_euc_realized_Ftot_and_cost_in_err_frac  2.336e+02  1.194e+01  19.554
## rsp_realized_FP_rate                       -2.271e+02  1.180e+01 -19.247
## rsp_realized_FN_rate                        1.008e+00  1.764e-01   5.713
## rsp_realized_Ftot_rate                             NA         NA      NA
## rsp_euc_realized_FP_and_cost_in_err_frac           NA         NA      NA
## rsp_euc_realized_FN_and_cost_in_err_frac           NA         NA      NA
##                                            Pr(>|t|)
## (Intercept)                                 < 2e-16 ***
## rsp_euc_realized_Ftot_and_cost_in_err_frac  < 2e-16 ***
## rsp_realized_FP_rate                        < 2e-16 ***
## rsp_realized_FN_rate                       1.19e-08 ***
## rsp_realized_Ftot_rate                            NA
## rsp_euc_realized_FP_and_cost_in_err_frac          NA
## rsp_euc_realized_FN_and_cost_in_err_frac          NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2024 on 3992 degrees of freedom
## Multiple R-squared:  0.4329, Adjusted R-squared:  0.4325
## F-statistic:  1016 on 3 and 3992 DF,  p-value: < 2.2e-16
```

```
plot (working_lm_fit)
```




Normal Q−Q
Standardized residuals
4
2
0
−2
−4
−2
0
2
Theoretical Quantiles
lm(rsr_COR_euc_out_err_frac ~ rsp_euc_realized_Ftot_and_cost_in_err_frac + ...
Scale−Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
2118
3484
991
0.2
0.4
0.6
0.8
1.0
Fitted values
lm(rsr_COR_euc_out_err_frac ~ rsp_euc_realized_Ftot_and_cost_in_err_frac + ...

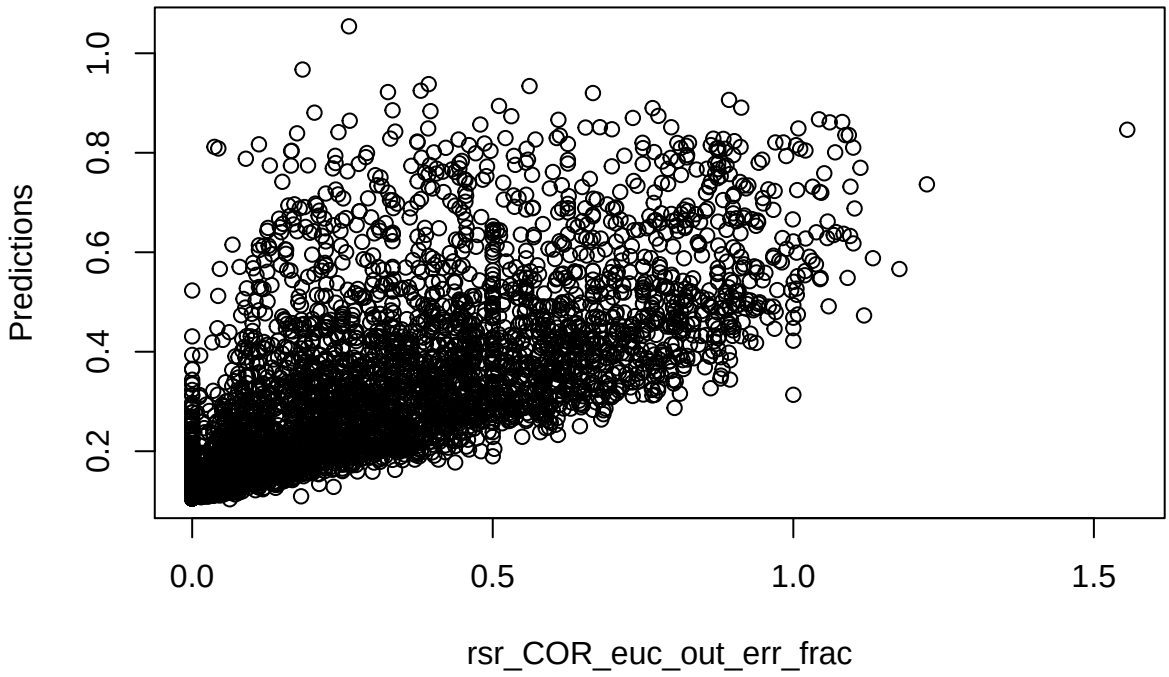


8.3.6 Plot Working COR OutErr vs. Fit Values from Cheating

```
pred_working_lm_fit =
    predict (working_lm_fit,
             working_df)
```

```
## Warning in predict.lm(working_lm_fit, working_df): prediction from a rank-
## deficient fit may be misleading
```

```
plot (working_df$rsr_COR_euc_out_err_frac,
      pred_working_lm_fit,
      xlab="rsr_COR_euc_out_err_frac", ylab="Predictions", main="Working COR OutErr vs. Fit Values from
```




```
# working_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df

working_not_huge_out_err_lm_fit =
    lm (rsr_COR_euc_out_err_frac ~ rsp_euc_realized_Ftot_and_cost_in_err_frac +
                                   rsp_realized_FP_rate +
                                   rsp_realized_FN_rate +
                                   rsp_realized_Ftot_rate +
                                   rsp_euc_realized_FP_and_cost_in_err_frac +
                                   rsp_euc_realized_FN_and_cost_in_err_frac,
        working_not_huge_out_err_df)

summary (working_not_huge_out_err_lm_fit)
plot (working_not_huge_out_err_lm_fit)
```

8.3.6.1 Cheating, using input errors and non-cheating problem characteristics

```
cat ("\n...  CHEATING:  Fit prediction using input error, graph measures, problem size  ...\n")
```

```
##
## ...  CHEATING:  Fit prediction using input error, graph measures, problem size  ...
```

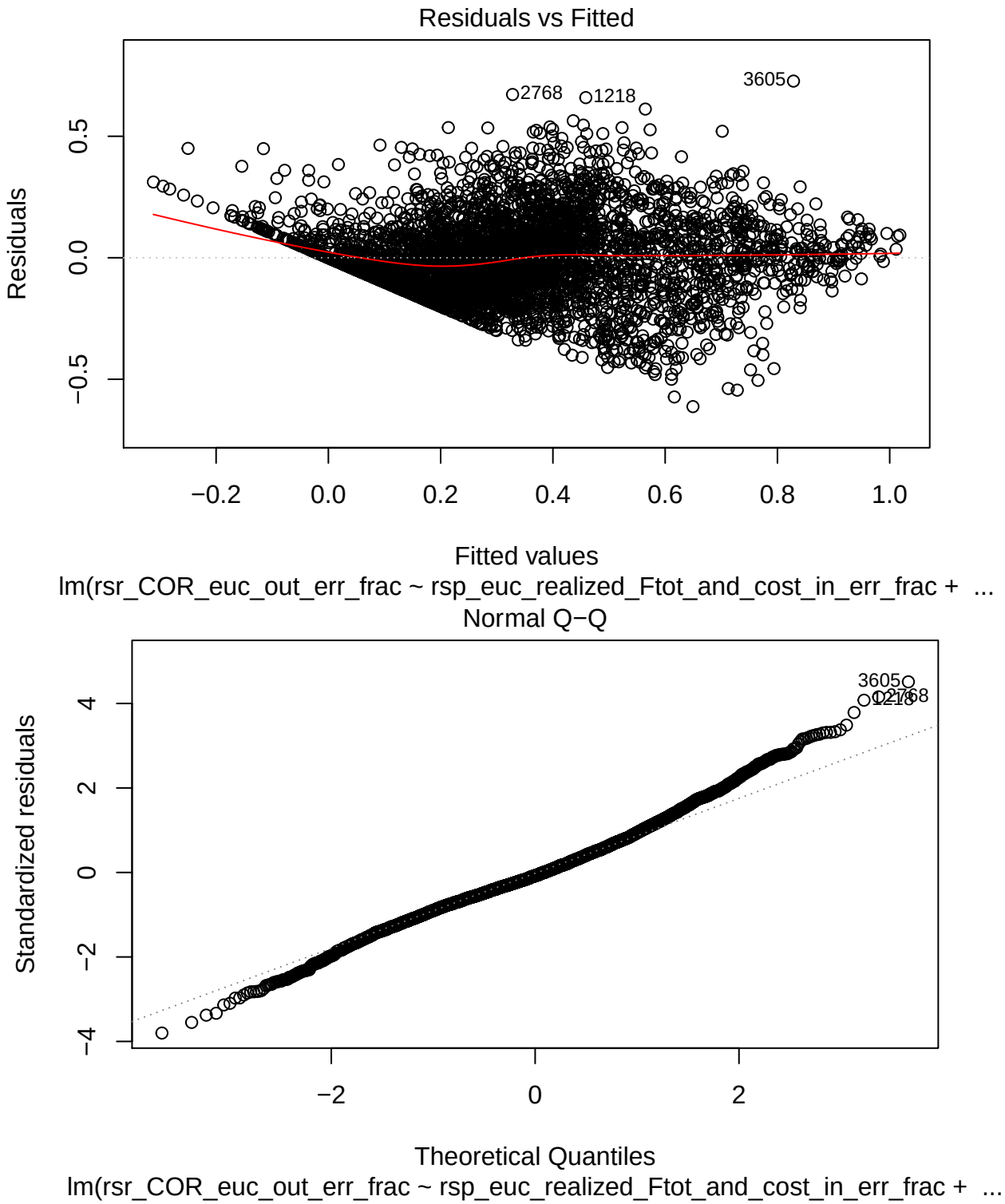
```
working_lm_fit = lm (rsr_COR_euc_out_err_frac ~ rsp_euc_realized_Ftot_and_cost_in_err_frac +
                                   rsp_realized_FP_rate +
                                   rsp_realized_FN_rate +
                                   rsp_realized_Ftot_rate +
                                   rsp_euc_realized_FP_and_cost_in_err_frac +
                                   rsp_euc_realized_FN_and_cost_in_err_frac +
                           ig_median_bottom_bg_redundancy + ig_median_top_bg_redundancy +
                           connectance +
```

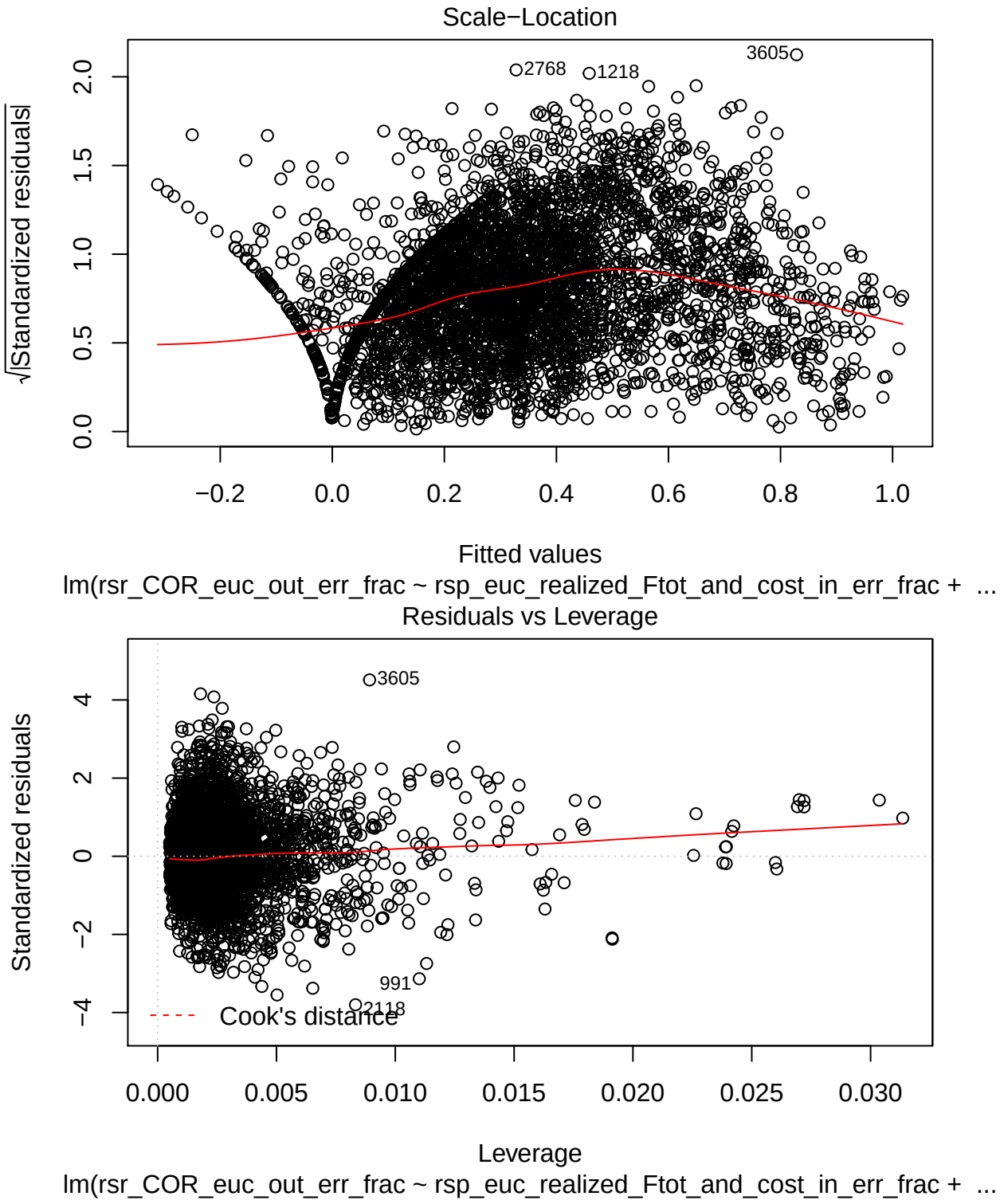
```
                              rsp_num_PUs + rsp_num_spp +
                                  rsp_num_spp_per_PU + sppPUprod,
                      working_df)

summary (working_lm_fit)
```

```
##
## Call:
## lm(formula = rsr_COR_euc_out_err_frac ~ rsp_euc_realized_Ftot_and_cost_in_err_frac +
##     rsp_realized_FP_rate + rsp_realized_FN_rate + rsp_realized_Ftot_rate +
##     rsp_euc_realized_FP_and_cost_in_err_frac + rsp_euc_realized_FN_and_cost_in_err_frac +
##     ig_median_bottom_bg_redundancy + ig_median_top_bg_redundancy +
##     connectance + rsp_num_PUs + rsp_num_spp + rsp_num_spp_per_PU +
##     sppPUprod, data = working_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61244 -0.09999 -0.01263  0.09340  0.72695
##
## Coefficients: (3 not defined because of singularities)
##                                           Estimate Std. Error t value
## (Intercept)                             -7.672e-01  5.749e-02 -13.344
## rsp_euc_realized_Ftot_and_cost_in_err_frac  2.452e+02  1.190e+01  20.606
## rsp_realized_FP_rate                    -2.595e+02  1.133e+01 -22.902
## rsp_realized_FN_rate                     1.056e+00  1.719e-01   6.144
## rsp_realized_Ftot_rate                          NA         NA      NA
## rsp_euc_realized_FP_and_cost_in_err_frac        NA         NA      NA
## rsp_euc_realized_FN_and_cost_in_err_frac        NA         NA      NA
## ig_median_bottom_bg_redundancy          -7.014e-01  6.661e-02 -10.530
## ig_median_top_bg_redundancy              1.529e+00  8.173e-02  18.706
## connectance                              1.214e+01  1.186e+00  10.238
## rsp_num_PUs                              1.762e-03  1.295e-04  13.610
## rsp_num_spp                              2.745e-05  1.558e-04   0.176
## rsp_num_spp_per_PU                       3.180e-01  4.286e-02   7.420
## sppPUprod                               -1.168e-06  1.357e-07  -8.602
##                                           Pr(>|t|)
## (Intercept)                                < 2e-16 ***
## rsp_euc_realized_Ftot_and_cost_in_err_frac  < 2e-16 ***
## rsp_realized_FP_rate                       < 2e-16 ***
## rsp_realized_FN_rate                      8.82e-10 ***
## rsp_realized_Ftot_rate                         NA
## rsp_euc_realized_FP_and_cost_in_err_frac       NA
## rsp_euc_realized_FN_and_cost_in_err_frac       NA
## ig_median_bottom_bg_redundancy             < 2e-16 ***
## ig_median_top_bg_redundancy                < 2e-16 ***
## connectance                                < 2e-16 ***
## rsp_num_PUs                                < 2e-16 ***
## rsp_num_spp                                  0.86
## rsp_num_spp_per_PU                        1.42e-13 ***
## sppPUprod                                  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1618 on 3985 degrees of freedom
```

```
## Multiple R-squared:  0.6383, Adjusted R-squared:  0.6374
## F-statistic: 703.4 on 10 and 3985 DF,  p-value: < 2.2e-16
```

```
plot (working_lm_fit)
```





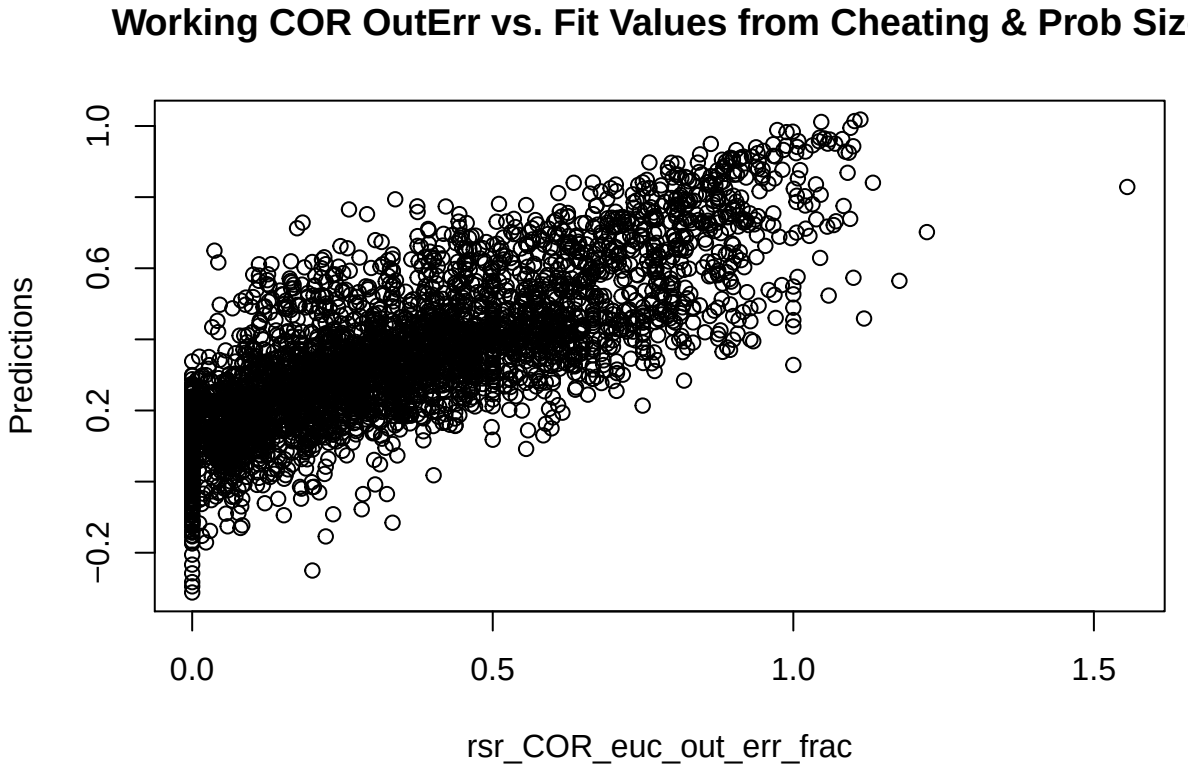
8.3.7 Plot Working COR OutErr vs. Fit Values from Cheating & Prob Size

```
pred_working_lm_fit =
    predict (working_lm_fit,
```

```
              working_df)
```

```
## Warning in predict.lm(working_lm_fit, working_df): prediction from a rank-
## deficient fit may be misleading
```

```
plot (working_df$rsr_COR_euc_out_err_frac,
      pred_working_lm_fit,
      xlab="rsr_COR_euc_out_err_frac", ylab="Predictions", main="Working COR OutErr vs. Fit Values from
```




```
# working_df %>%
#     filter (rsr_COR_euc_out_err_frac <= 1) -> not_huge_out_err_df

working_not_huge_out_err_lm_fit =
    lm (rsr_COR_euc_out_err_frac ~ rsp_euc_realized_Ftot_and_cost_in_err_frac +
                                   rsp_realized_FP_rate +
                                   rsp_realized_FN_rate +
                                   rsp_realized_Ftot_rate +
                                   rsp_euc_realized_FP_and_cost_in_err_frac +
                                   rsp_euc_realized_FN_and_cost_in_err_frac +
                           ig_median_bottom_bg_redundancy + ig_median_top_bg_redundancy +
                           connectance +
                               rsp_num_PUs + rsp_num_spp +
                                   rsp_num_spp_per_PU + sppPUprod,
        working_not_huge_out_err_df)

summary (working_not_huge_out_err_lm_fit)
plot (working_not_huge_out_err_lm_fit)
```

8.4 Look at residuals using `olsrr` package

Copied from https://cran.r-project.org/web/packages/olsrr/vignettes/residual_diagnostics.html

Residual Diagnostics Introduction

olsrr offers tools for detecting violation of standard regression assumptions. Here we take a look at residual diagnostics. The standard regression assumptions include the following about residuals/errors:

```
The error has a normal distribution (normality assumption).
The errors have mean zero.
The errors have same but unknown variance (homoscedasticity assumption).
The error are independent of each other (independent errors assumption).
```

```
library (olsrr)
```

```
## Warning: package 'olsrr' was built under R version 3.4.4
```

```
##
## Attaching package: 'olsrr'
```

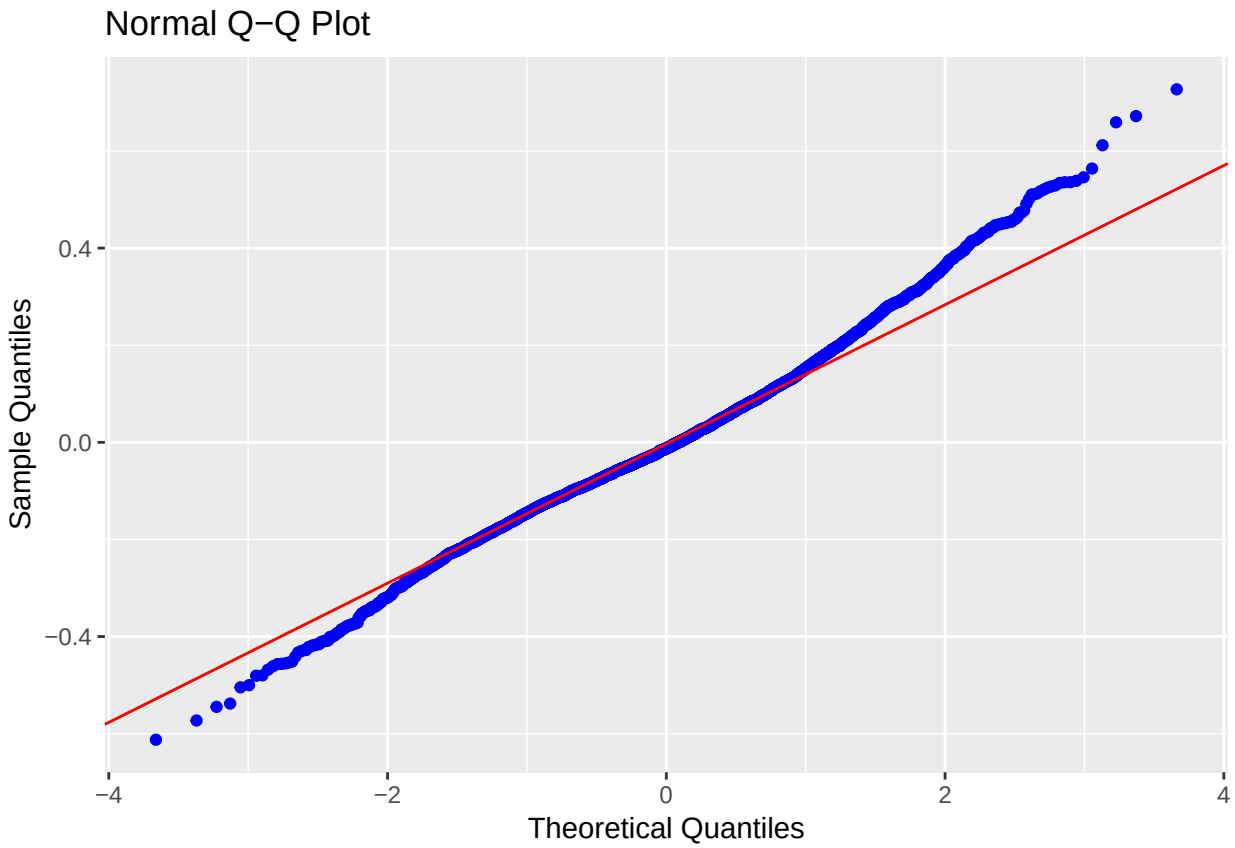
```
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
#model = working_not_huge_out_err_lm_fit
#####model = working_not_huge_out_err_lm_fit__using_graph_and_prob_size
#####model = working_not_huge_out_err_lm_fit__using_graph_and_prob_size
model = working_lm_fit
```

8.4.1 Residual QQ Plot

Graph for detecting violation of normality assumption.

```
ols_plot_resid_qq (model)
```



8.4.2 Residual Normality Test

Test for detecting violation of normality assumption.

```
ols_test_normality (model)
```

```
## Warning in ks.test(y, "pnorm", mean(y), sd(y)): ties should not be present
## for the Kolmogorov-Smirnov test
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9899         0.0000
## Kolmogorov-Smirnov        0.0402         0.0000
## Cramer-von Mises        955.9591         0.2038
## Anderson-Darling         12.3203         0.0000
## -----------------------------------------------
```

8.4.3 Correlation between observed residuals and expected residuals under normality.

```
ols_test_correlation (model)
```

```
## [1] 0.9948668
```

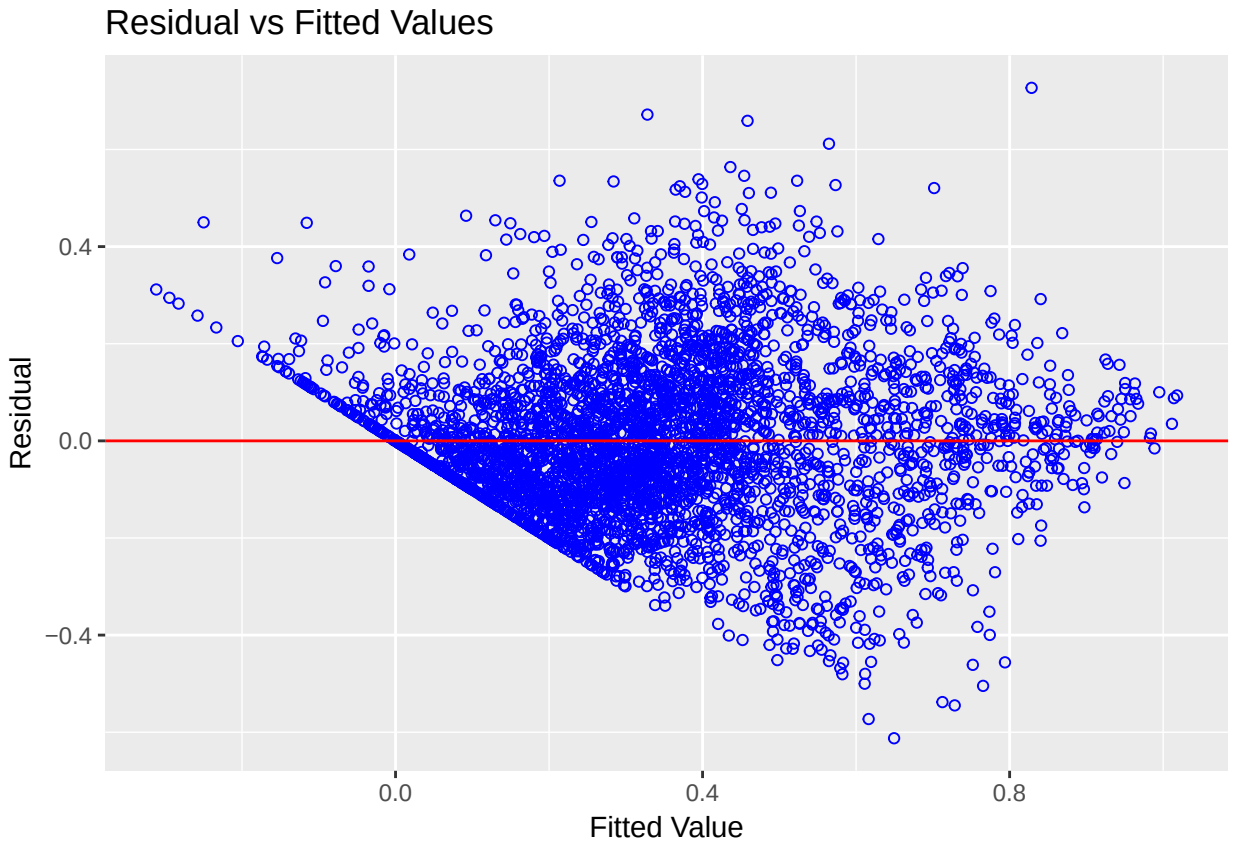
8.4.4 Residual vs Fitted Values Plot

It is a scatter plot of residuals on the y axis and fitted values on the x axis to detect non-linearity, unequal error variances, and outliers.

Characteristics of a well behaved residual vs fitted plot:

```
The residuals spread randomly around the 0 line indicating that the relationship is linear.
The residuals form an approximate horizontal band around the 0 line indicating homogeneity of error vari
No one residual is visibly away from the random pattern of the residuals indicating that there are no ou
```

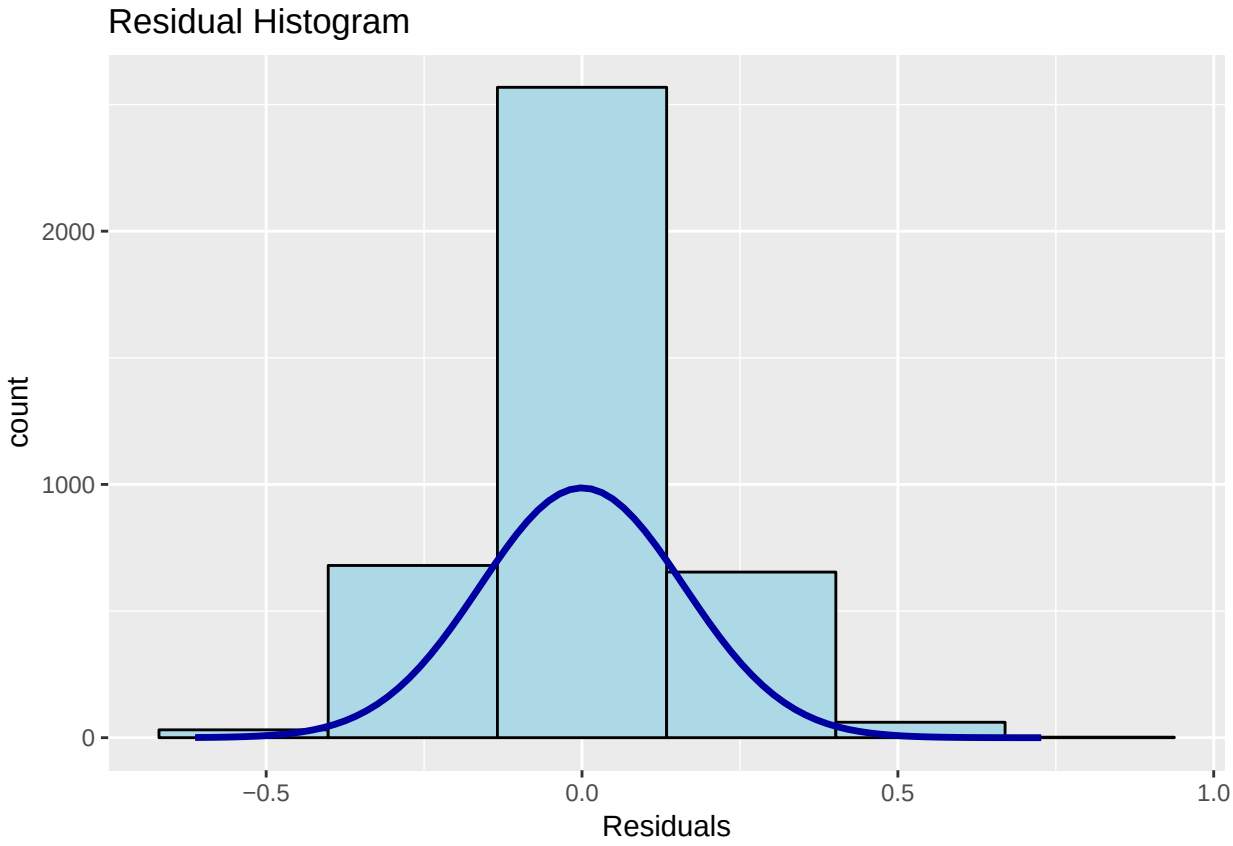
```
ols_plot_resid_fit (model)
```



8.4.5 Residual Histogram

Histogram of residuals for detecting violation of normality assumption.

```
ols_plot_resid_hist (model)
```



8.5 Look at residuals using `car` package

Copied from https://www.statmethods.net/stats/rdiagnostics.html

Regression Diagnostics

An excellent review of regression diagnostics is provided in John Fox's aptly named Overview of Regression Diagnostics (http://socserv.socsci.mcmaster.ca/jfox/Courses/Brazil-2009/index.html). Dr. Fox's `car` package provides advanced utilities for regression modeling.

```
library (car)
```

```
## Warning: package 'car' was built under R version 3.4.4
## Loading required package: carData
## Warning: package 'carData' was built under R version 3.4.4
##
## Attaching package: 'car'
## The following object is masked from 'package:dplyr':
##
##     recode
## The following object is masked from 'package:purrr':
##
##     some
```

```
#fit = working_not_huge_out_err_lm_fit
#####fit = working_not_huge_out_err_lm_fit__using_graph_and_prob_size
fit = working_lm_fit
```

8.5.1 Outliers

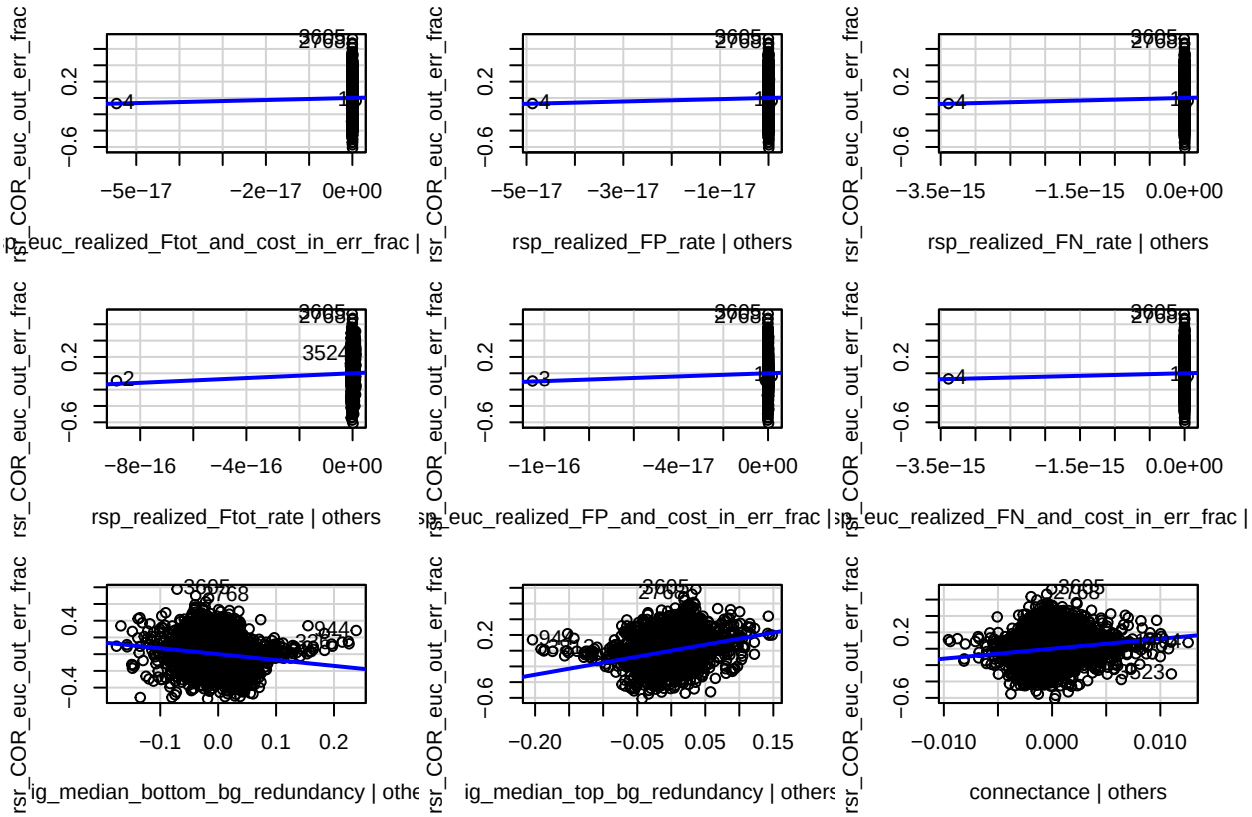
```
    # Assessing Outliers
outlierTest (fit) # Bonferonni p-value for most extreme obs
qqPlot (fit, main="QQ Plot") #qq plot for studentized resid

leveragePlots (fit) # leverage plots
#  THIS FAILS:
#leveragePlots (fit) # leverage plots
#  ERROR OUTPUT:
#Error in L %*% V : non-conformable arguments
#4.solve(L %*% V %*% t(L))
#3.leveragePlot.lm(model, term, main = "", ...)
#2.leveragePlot(model, term, main = "", ...)
#1.leveragePlots(fit)
```

8.5.2 Influential Observations

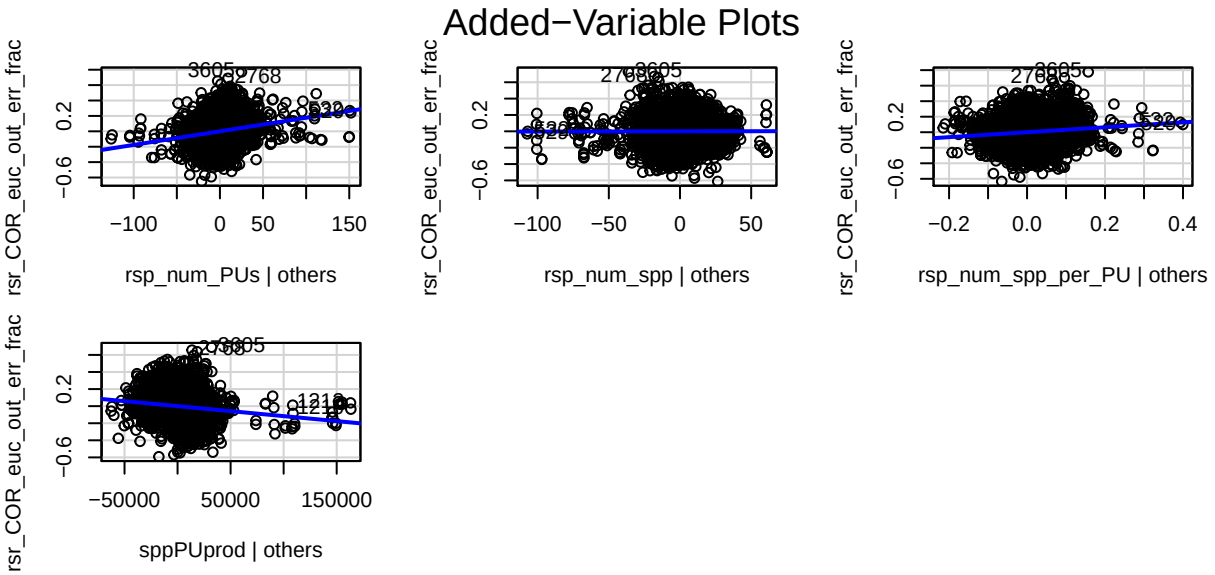
```
    # added variable plots
#av.Plots (fit)
avPlots (fit)
```

```
## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear

## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear

## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear

## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear

## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear

## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear

## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear

## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear

## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear
```

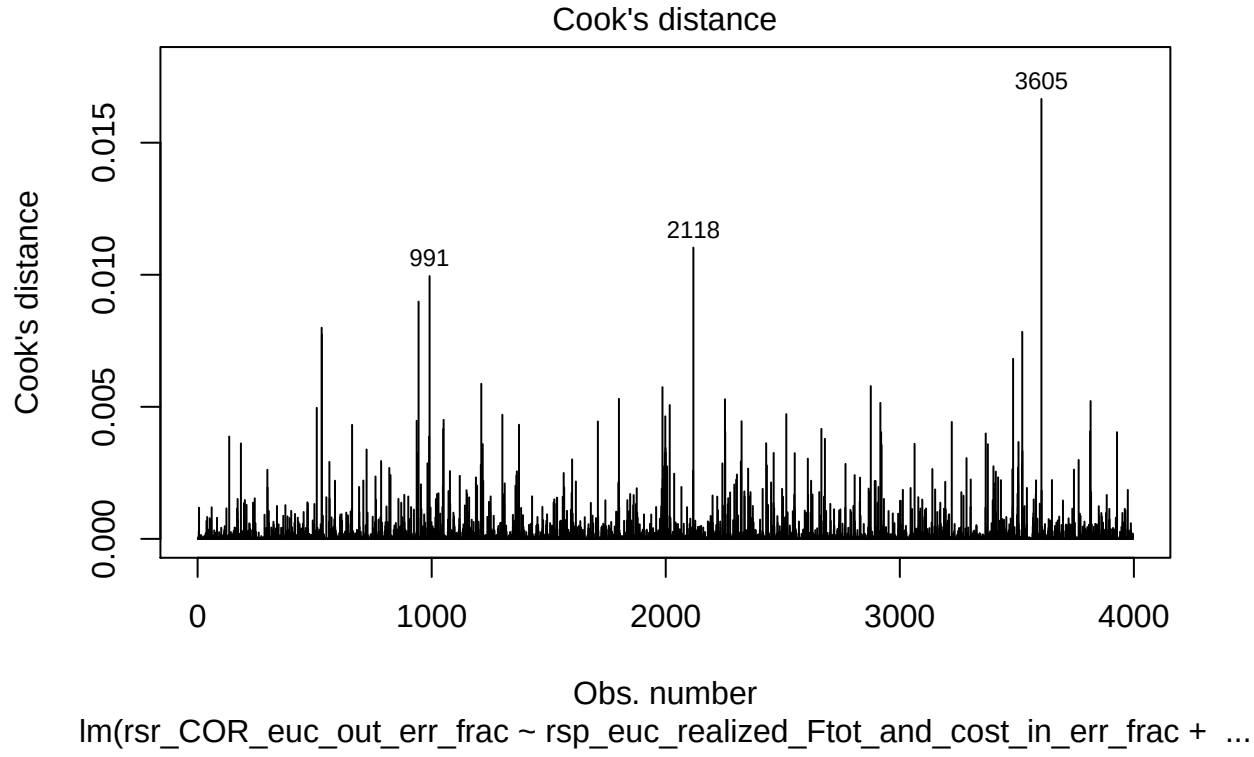
```
## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear
```




```
## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear

## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear

## Warning in lsfit(mod.mat[, -var], cbind(mod.mat[, var], response), wt =
## wt, : 'X' matrix was collinear
```




```
#    THIS SECTION IS RELATED TO THE mtcars DATASET, NOT MY DATA.
#    NEED TO FIGURE OUT IF/HOW IT TRANSLATES TO MY DATA.
#
#     # Cook's D plot
#     # identify D values > 4/(n-k-1)

# cutoff <- 4 / ((nrow (mtcars)                                  - length (fit$coefficients) - 2))
#####  cutoff <- 4 / ((nrow (working_not_huge_out_err_df) - length (fit$coefficients) - 2))
  cutoff <- 4 / ((nrow (working_df) - length (fit$coefficients) - 2))

plot(fit, which=4, cook.levels=cutoff)
```



That plot marks rows 2094, 3032, and 3033. Other plots also mention 2092.

```
#glimpse (working_not_huge_out_err_df [c(2092, 2094, 3032, 3033), ])

#####print (working_not_huge_out_err_df [c(2092, 2094, 3032, 3033), ])
```

This influencePlot call gives the same error message 7 times. I think that the format of the call may be out of date. The message it gives is:

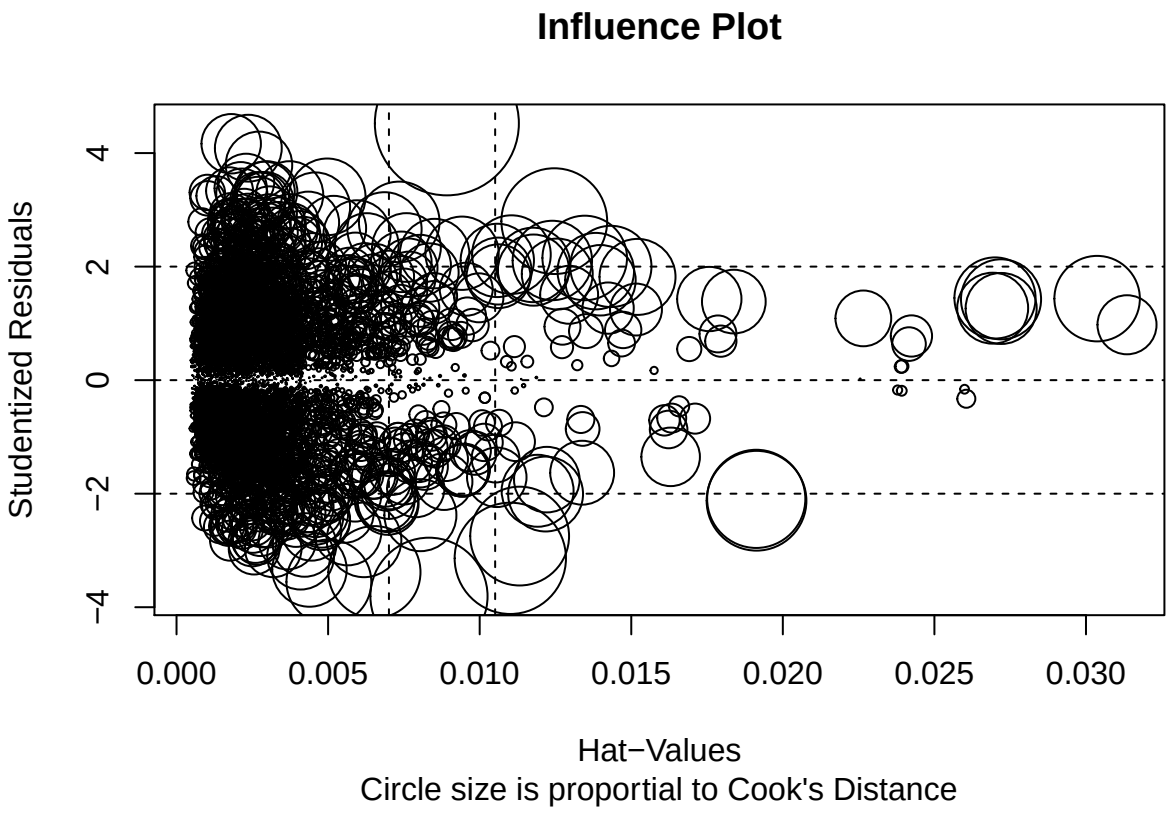
```
"id.method" is not a graphical parameter
```

Looking at the help page for `influencePlot` shows a different way to invoke the "identify" option:

```
id
settings for labelling points; see link{showLabels} for details. To omit point labelling, set id=FALSE;
...
Examples

...
## Not run:
influencePlot(lm(prestige ~ income + education, data=Duncan),
    id=list(method="identify"))

## End(Not run)
```

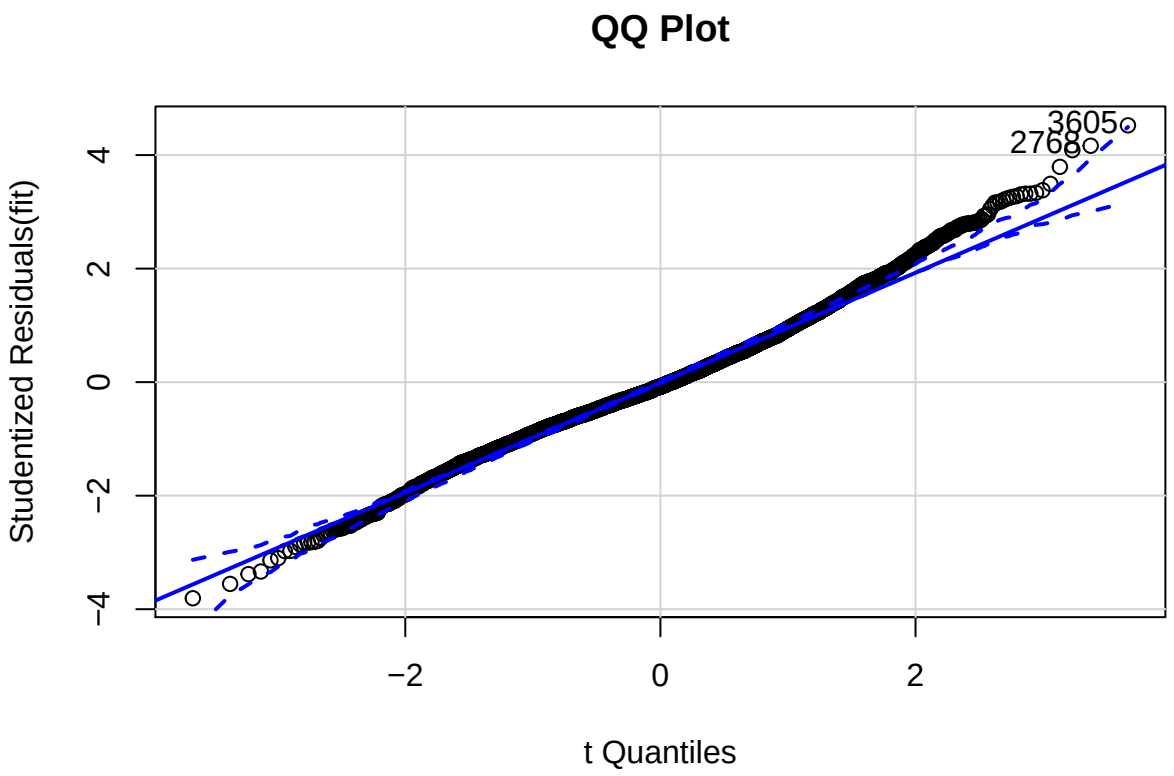
So, modifying the influencePlot() call to match that gives:

```
    # Influence Plot
influencePlot (fit,

                      #id.method="identify",
                      id=list (method="identify"),

               main="Influence Plot", sub="Circle size is proportial to Cook's Distance" )
```



8.5.3 Non-normality

```
    # Normality of Residuals
    # qq plot for studentized resid
qqPlot (fit, main="QQ Plot")
```




```
## [1] 2768 3605
```

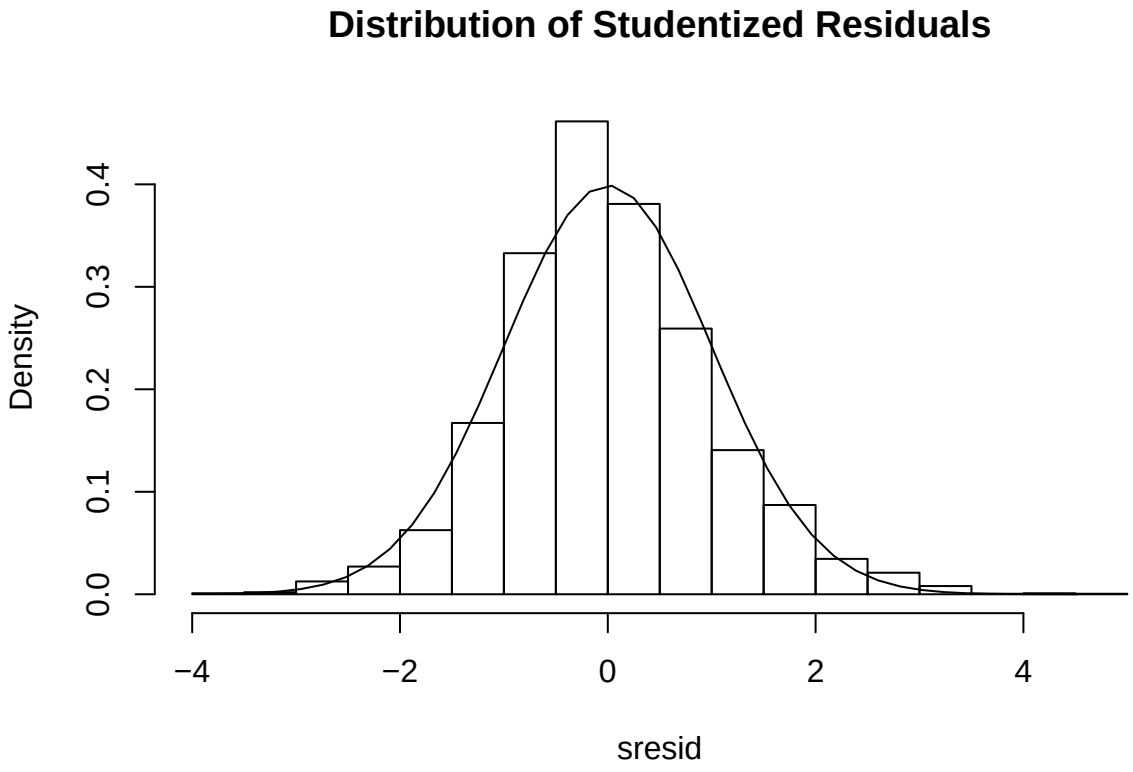
```
    # distribution of studentized residuals
library (MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:olsrr':
##
##     cement
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
sresid <- studres(fit)
hist (sresid, freq=FALSE, main="Distribution of Studentized Residuals")
xfit <- seq (min (sresid), max (sresid), length=40)
yfit <- dnorm (xfit)
lines (xfit, yfit)
```



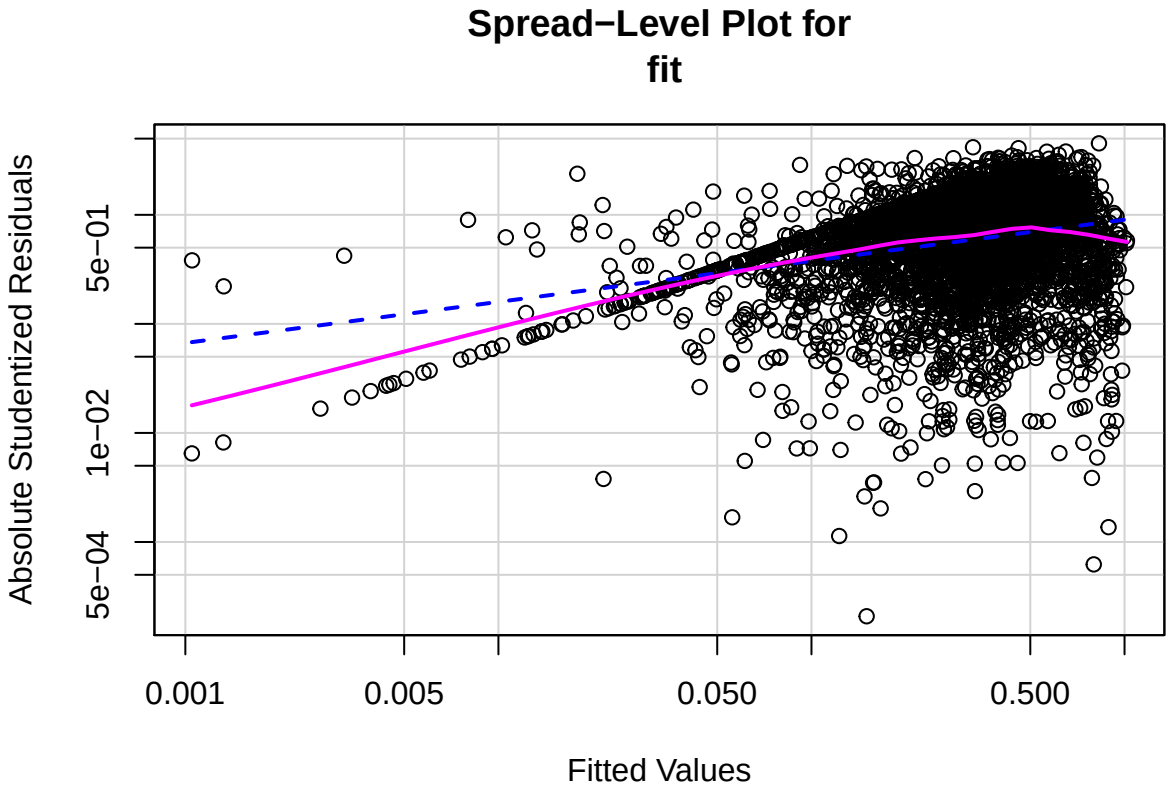
8.5.4 Non-constant Error Variance

```
    # Evaluate homoscedasticity
    # non-constant error variance test
ncvTest (fit)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 171.9829    Df = 1     p = 2.729644e-39
```

```
    # plot studentized residuals vs. fitted values
spreadLevelPlot (fit)
```

```
## Warning in spreadLevelPlot.lm(fit):
## 146 negative fitted values removed
```




```
##
## Suggested power transformation:  0.6233275
```

8.5.5 Multi-collinearity

Not running this because it currently gives the following error message:

```
Error in vif.default(fit) : there are aliased coefficients in the model
```

```
    # Evaluate Collinearity
vif (fit)               # variance inflation factors
sqrt (vif (fit)) > 2    # problem?
```

8.5.6 Nonlinearity

Not running this because it currently gives the following error message:

```
Error in df.terms.default(model, var) :
  Model has aliased term(s); df ambiguous.
```

```
    # Evaluate Nonlinearity
    # component + residual plot
crPlots (fit)
```

Not running this because it currently gives the following error message:

```
no non-missing arguments to min; returning Infno non-missing arguments to max; returning -Inf
Hit <Return> to see next plot:
Error in plot.window(...) : need finite 'ylim' values
```

```
    # Ceres plots
ceresPlots (fit)
```

8.5.7 Non-independence of Errors

```
    # Test for Autocorrelated Errors
durbinWatsonTest(fit)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.07558579      1.848382       0
##  Alternative hypothesis: rho != 0
```

8.5.8 Additional Diagnostic Help

The gvlma( ) function in the gvlma package, performs a global validation of linear model assumptions as well separate evaluations of skewness, kurtosis, and heteroscedasticity.

Not running this because it currently gives the following error message:

```
Error in solve.default(sigwhat) :
  Lapack routine dgesv: system is exactly singular: U[12,12] = 0
```

```
    # Global test of model assumptions
library (gvlma)

gvmodel <- gvlma (fit)
summary (gvmodel)
```

8.5.9 Plot the last fit, first with a confidence interval and then with a prediction interval.

This code is derived from: https://stackoverflow.com/questions/44865508/using-ggplot2-to-plot-an-already-existing-linear-mod

For plotting,

- the x-axis should be an input error, e.g., to start with, the total input error.
- the y-axis should be an output error, e.g., to start with, the total output error.

```
# # add 'fit', 'lwr', and 'upr' columns to dataframe (generated by predict)
# cars.predict <- cbind(cars, predict(cars.model, interval = 'confidence'))
```

9 Does Easy vs. Hard matter?

Since all existing papers use easy problems (small toy problems or random problems), does that matter?

9.0.0.1 Are descriptive results different?

```
cat ("\n...  Break down by Easy vs. Hard  ...\n")
```

```
##
## ...  Break down by Easy vs. Hard  ...
```

9.0.0.2 Are predictive results different?

Can use the Easy/Hard labels with the same facetting code I had earlier for Base vs Wrap plots.

- If I train on Easy, how well do I do predicting on Hard?
- If I train on Hard, how well do I do predicting on Easy?
- If I train on mixed, how well do I do predicting on mixed?

```
cat ("\n...  Predict using different train/test sources  ...\n")
```

```
##
## ...  Predict using different train/test sources  ...
```