R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

```r
library(tidyverse)
```

```
## -- Attaching packages -----------------------------------------------------------------

## v ggplot2 3.2.0      v purrr   0.3.2
## v tibble  2.1.3      v dplyr   0.8.3
## v tidyr   0.8.3      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0

## -- Conflicts --------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

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

```r
library(grid)
```

Some plotting functions

```r
plot_mriqc_metric <- function(mriqc,metric){

  return (ggplot(data=mriqc, mapping = aes_string(x="subj", y=metric)) + geom_boxplot(outlier.shape=NA,

}


grid_arrange_shared_legend <- function(...) {
    plots <- list(...)
    g <- ggplotGrob(plots[[1]] + theme(legend.position="bottom"))$grobs
    legend <- g[[which(sapply(g, function(x) x$name) == "guide-box")]]
    lheight <- sum(legend$height)

  grid.arrange(arrangeGrob(grobs=lapply(plots, function(x)
      x + theme(legend.position="none",aspect.ratio=0.6))),
```

```
    legend,
    ncol = 1,
    heights = unit.c(unit(1, "npc") - lheight, lheight),
    widths = unit(0.5, "npc")


    )


}
```

Read in the functional data:

```
func<- read.csv('/projects/gherman/SPINS_human_phantoms/group_bold.tsv', sep='\t') %>%
  mutate(subj = substr(bids_name,1, 6), year = substr(bids_name, 8, 13), site=substr(bids_name, 14, 16)
```

```
mriqc=func
metric="tsnr"


fillz=c(NA, "black",NA,"black",NA,"black")

names(fillz)= c("CMH", "CMP", "MRC" ,"MRP", "ZHH" ,"ZHP")


ggplot(data=mriqc, mapping = aes_string(x="subj", y=metric)) + geom_boxplot(outlier.shape=NA, width=0.8
```
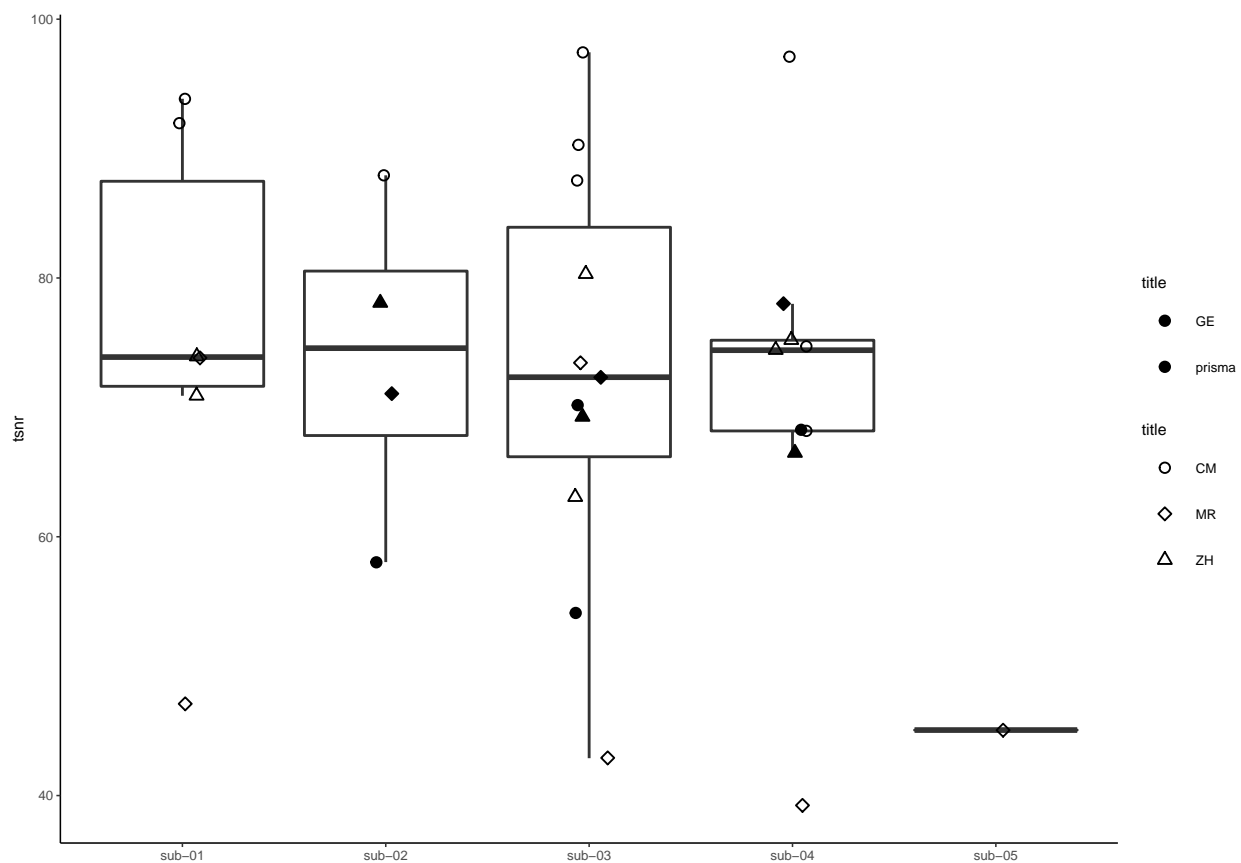


```
mriqc=func
metric="tsnr"
```
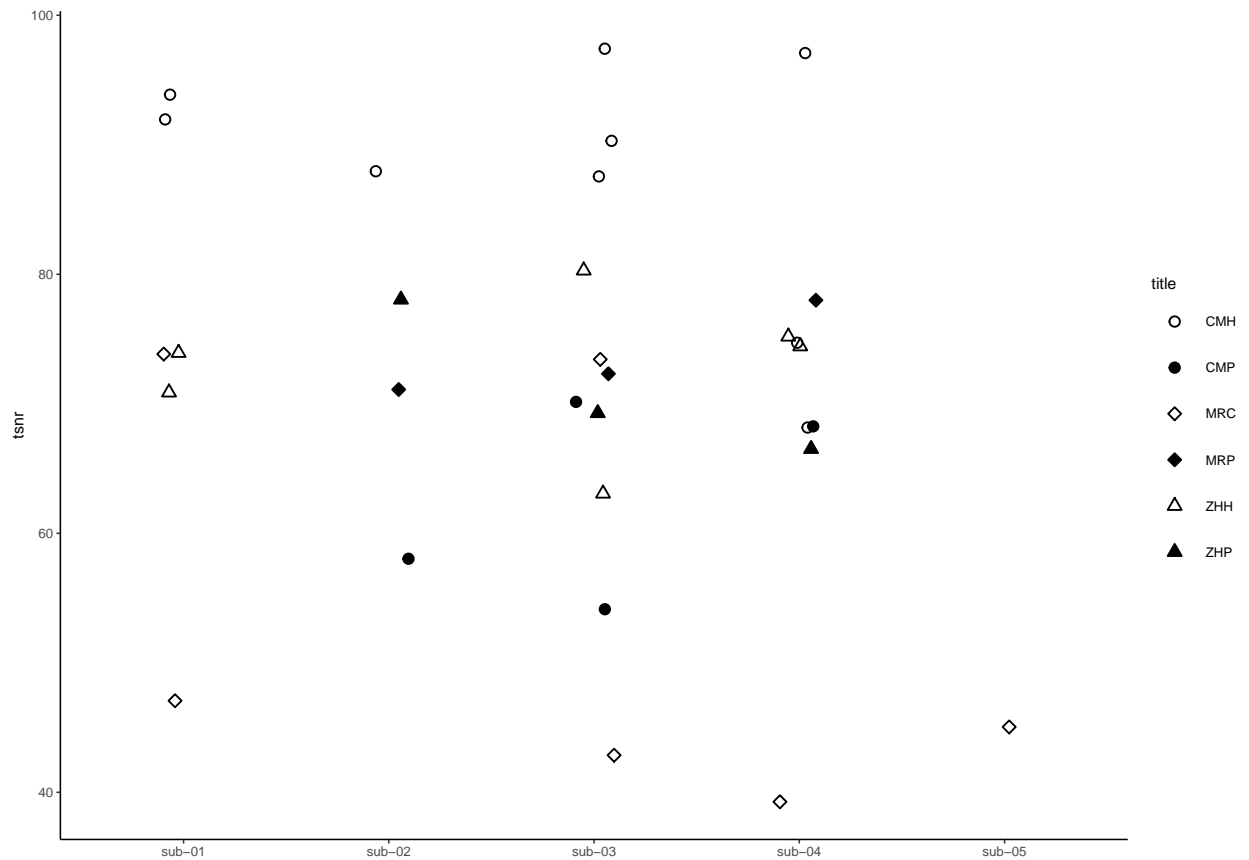
```r
shapez=c(21,23,24,21,23,24)
names(shapez)=c("CMH","MRP","ZHH","CMP","MRC","ZHP")



fillz=c(NA,"black",NA,"black",NA,"black")
names(fillz)=c("CMH","MRP","ZHH","CMP","MRC","ZHP")


ggplot(data=mriqc, mapping = aes_string(x="subj", y=metric)) + geom_jitter(width=0.1, aes( fill=site, sh
```


Plot some fmri metrics

```r
tsnr <- plot_mriqc_metric(func,"tsnr")

snr <- plot_mriqc_metric(func, "snr")

fd_mean <- plot_mriqc_metric(func,"fd_mean")

fd_perc <- plot_mriqc_metric(func, "fd_perc")

#gsr <- plot_mriqc_metric("gsr")

efc <- plot_mriqc_metric(func, "efc")

aor <- plot_mriqc_metric(func, "aor")
```
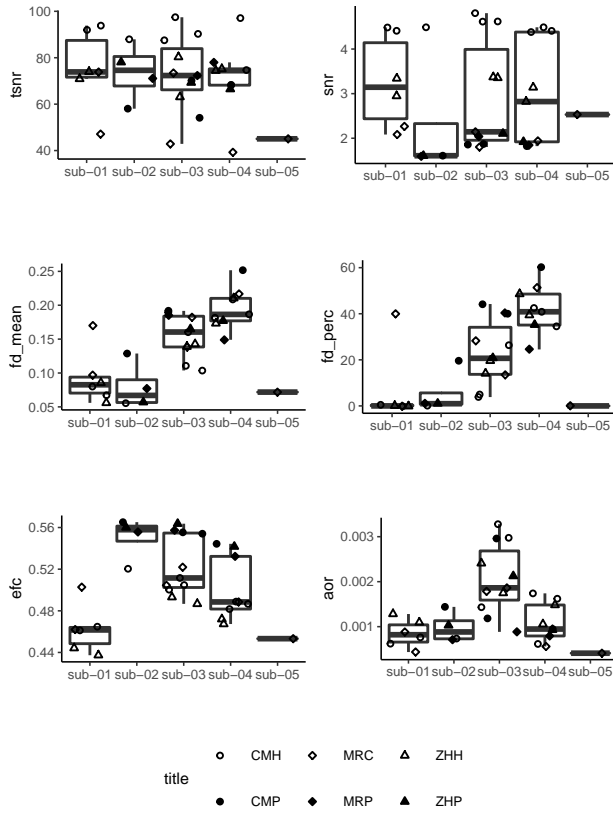
```
plt <- grid_arrange_shared_legend(tsnr, snr, fd_mean, fd_perc, efc, aor)
```


Read in the T1 data:

```
T1 <- read.csv('/projects/gherman/SPINS_human_phantoms/group_T1w.tsv', sep='\t') %>%
  mutate(subj = substr(bids_name,1, 6), year = substr(bids_name, 8, 13), site=substr(bids_name, 14, 16))
```

```
cnr <- plot_mriqc_metric(T1,"cnr")

snr_total <- plot_mriqc_metric(T1, "snr_total")

snrd_total <- plot_mriqc_metric(T1, "snrd_total")


fwhm_avg <- plot_mriqc_metric(T1, "fwhm_avg")

cjv <- plot_mriqc_metric(T1, "cjv")

efc <- plot_mriqc_metric(T1, "efc")

grid_arrange_shared_legend(cnr, snr_total,snrd_total, fwhm_avg, cjv, efc)
```
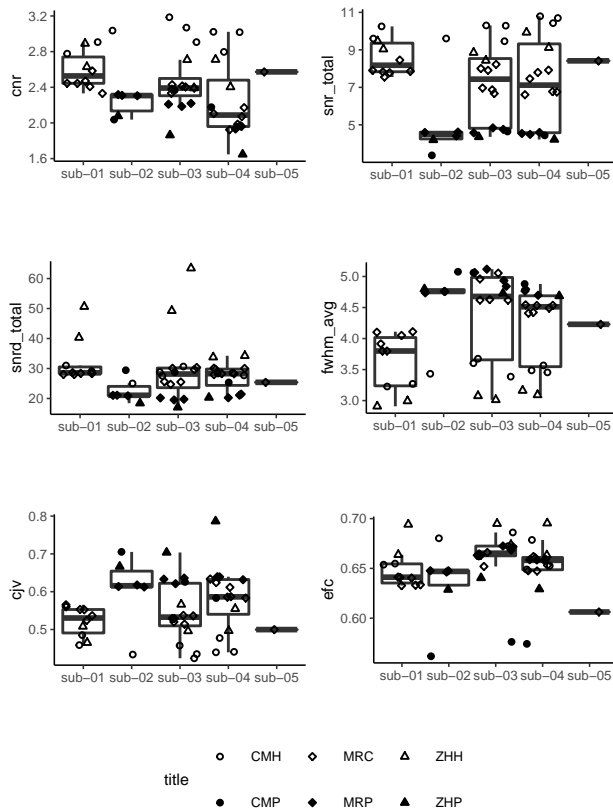
Read in the T2 data:

```r
T2 <- read.csv('/projects/gherman/SPINS_human_phantoms/group_T2w.tsv', sep='\t') %>%
  mutate(subj = substr(bids_name,1, 6), year = substr(bids_name, 8, 13), site=substr(bids_name, 14, 16))

cnr <- plot_mriqc_metric(T2,"cnr")

snr_total <- plot_mriqc_metric(T2, "snr_total")

snrd_total <- plot_mriqc_metric(T2, "snrd_total")


fwhm_avg <- plot_mriqc_metric(T2, "fwhm_avg")

cjv <- plot_mriqc_metric(T2, "cjv")

efc <- plot_mriqc_metric(T2, "efc")

grid_arrange_shared_legend(cnr, snr_total,snrd_total, fwhm_avg, cjv, efc)
```
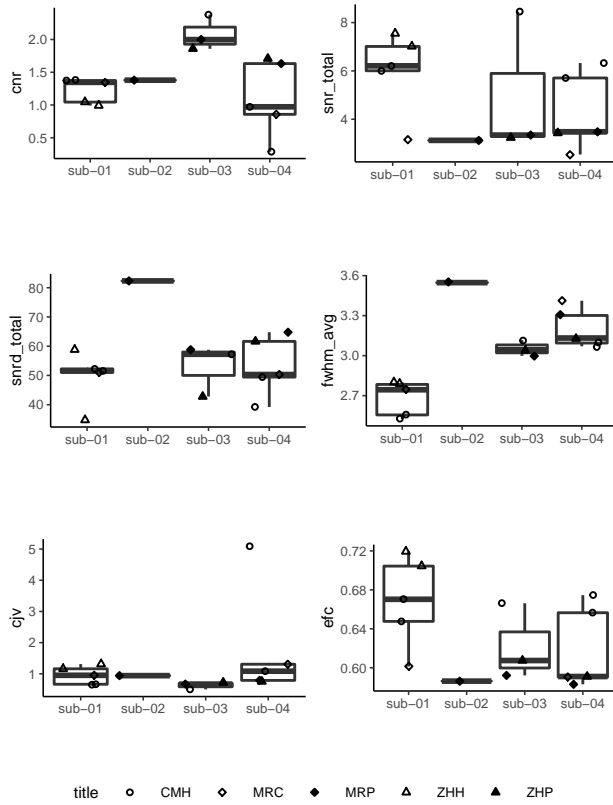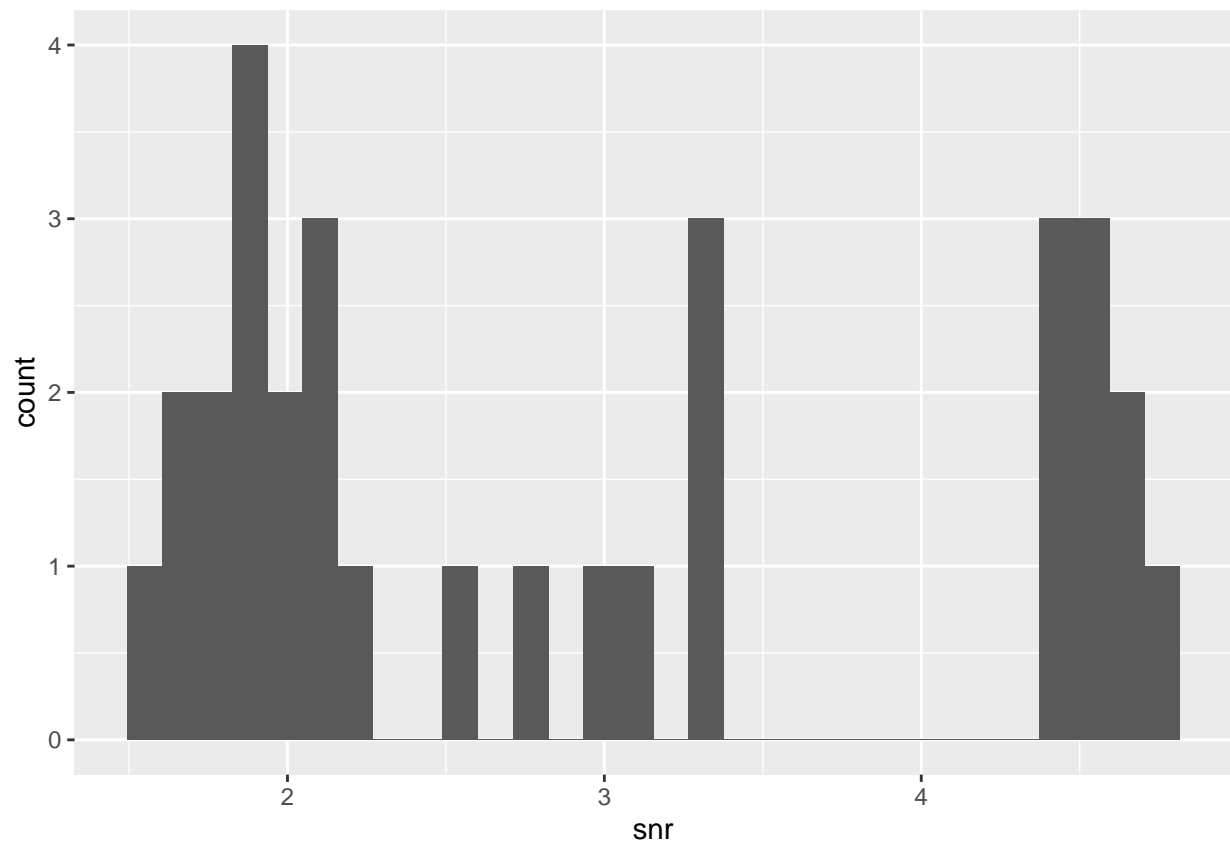
ewwwwwwwwwwww a very ugly histogram lol

```
ggplot(data=func)+ geom_histogram(mapping=aes(x=snr ), stat="bin") # geom_bar(mapping=aes(x=snr, y=..pr
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Should run the DTI pipelines on the human phantom data once they become available and we decide on the kind of metrics to use.