

# number estimates

Laura

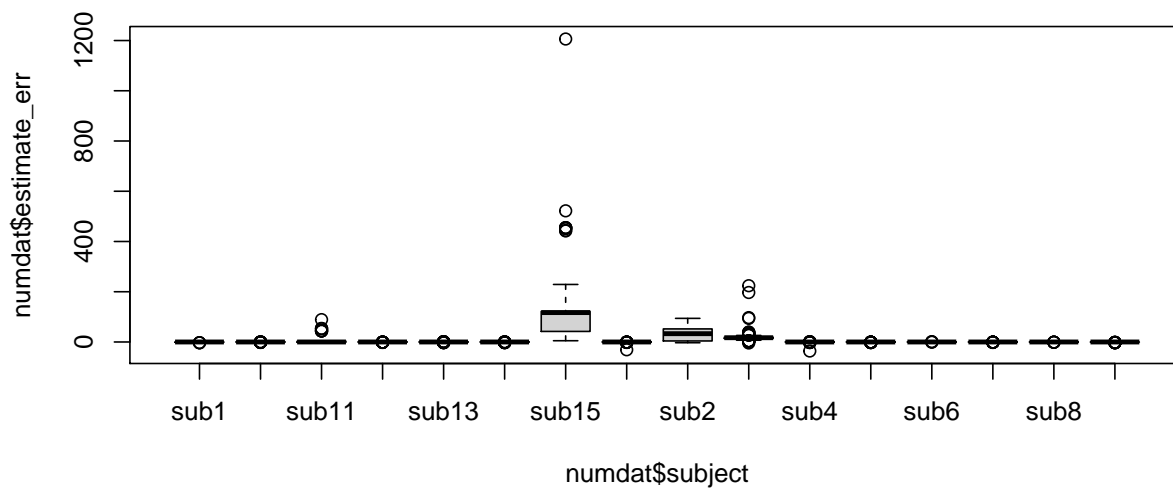
4/21/2021

## Error in number estimates by subject

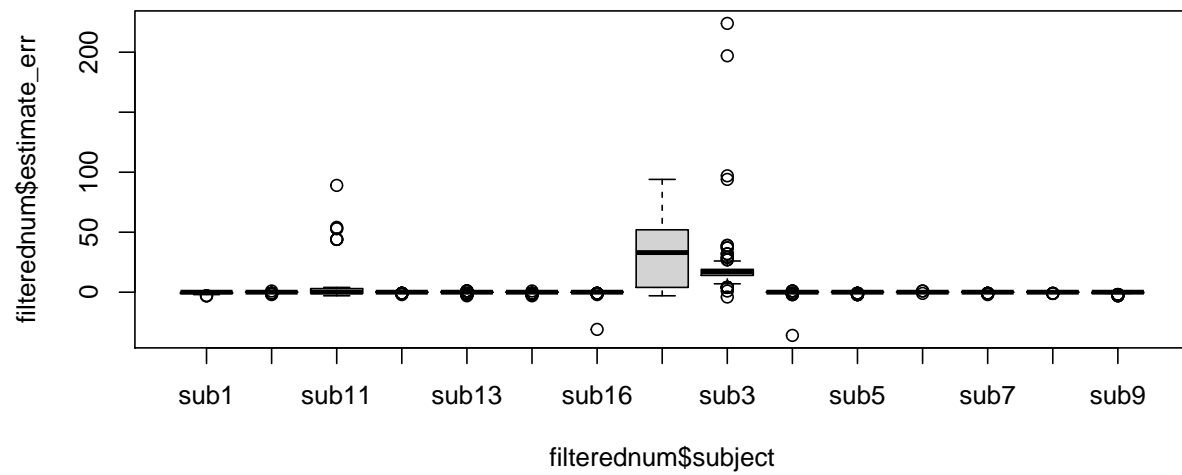
```
numdat <- read.csv("numdat.csv")

numdat <- numdat %>%
  mutate(iscorrect = ifelse(stim_number == subject_estimate, 1, 0))

##look at boxplots of all 16 subjects
boxplot(numdat$estimate_err ~ numdat$subject)
```

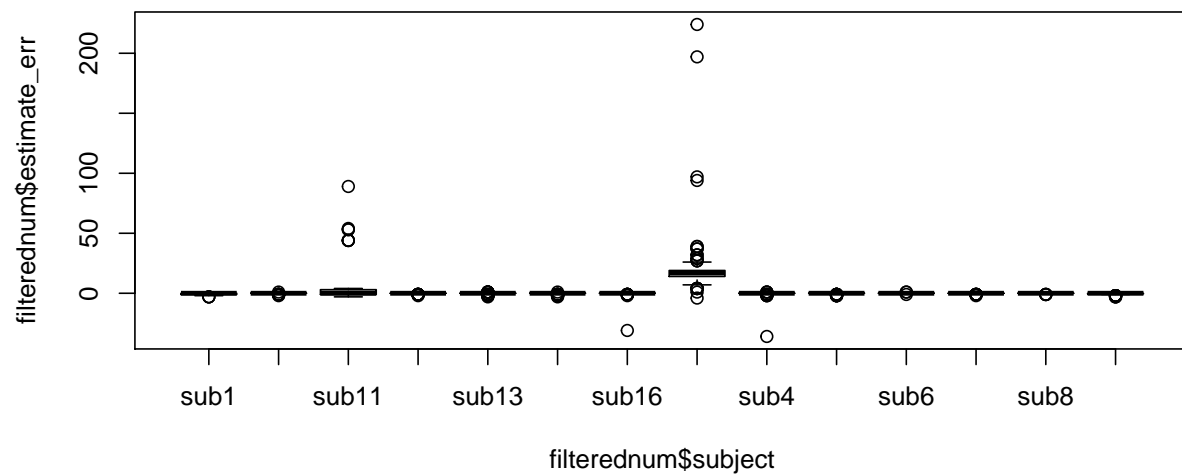


```
##take out sub15
filterednum <- numdat %>% dplyr::filter(subject != "sub15")
boxplot(filterednum$estimate_err ~ filterednum$subject)
```



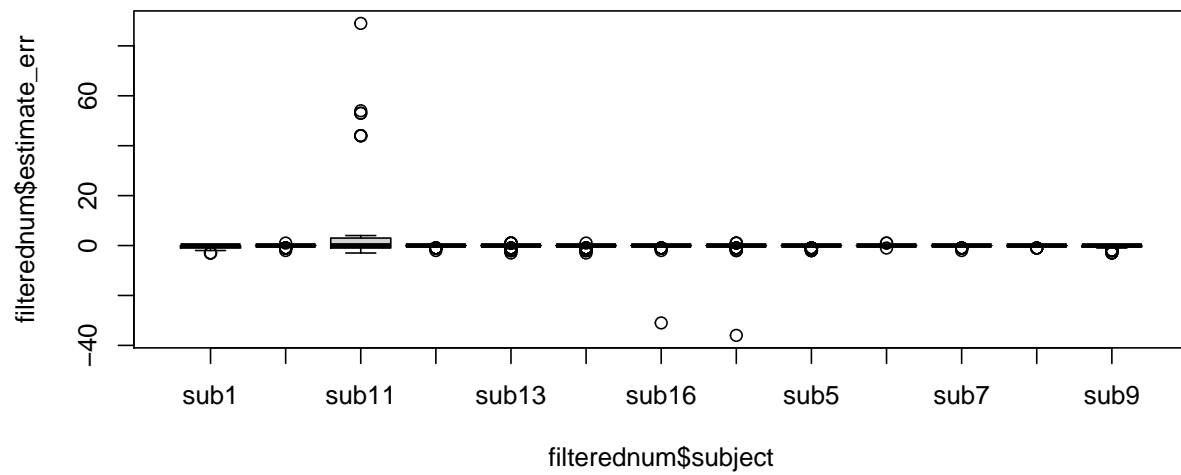
```
##take out sub2
```

```
filterednum <- filterednum %>% dplyr::filter(subject != "sub2")
boxplot(filterednum$estimate_err ~ filterednum$subject)
```

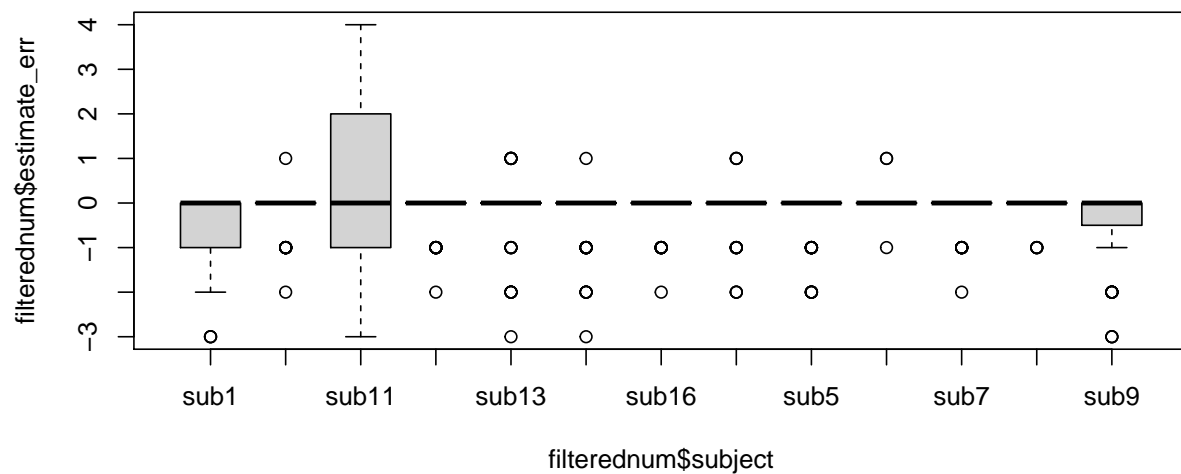


```
##take out sub3
```

```
filterednum <- filterednum %>% dplyr::filter(subject != "sub3")
boxplot(filterednum$estimate_err ~ filterednum$subject)
```



```
##now remove individual estimates with |errors| > 12 (assuming people just messed up occasionally)
filterednum <- filterednum %>% dplyr::filter(abs(estimate_err) < 12)
boxplot(filterednum$estimate_err ~ filterednum$subject)
```



```
# & subject != "sub2"& subject != "sub3") %>%
# filter(abs(estimate_err) < 12)
```

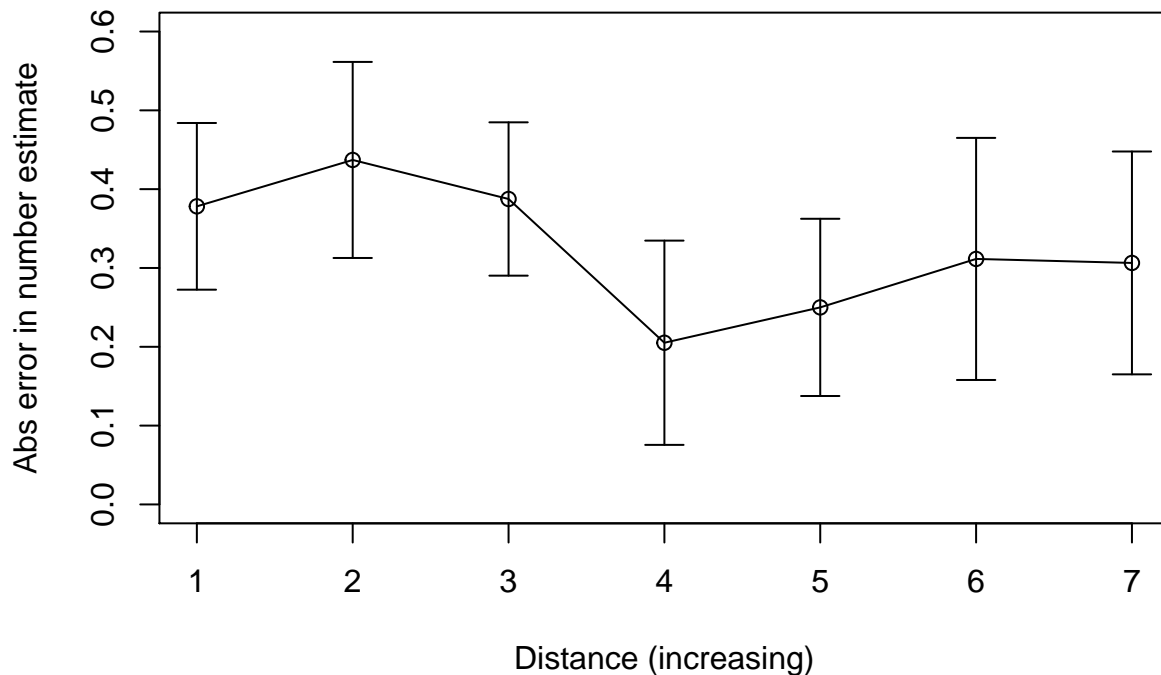
Seems reasonable. Now look by condition and distance

```
abs_err_means <- filterednum %>%
  select(subject, condition, stim_distance, estimate_err) %>%
  mutate(abs_err = abs(estimate_err)) %>%
  pivot_wider(names_from = stim_distance, values_from = abs_err, values_fn = mean, id_cols = c(subject,
```

```
dplyr::select(subject, condition, `15`, `25`, `50`, `75`, `100`, `150`, `200`)
print(abs_err_means)
```

```
## # A tibble: 13 x 9
##   subject condition  `15`  `25`  `50`  `75`  `100`  `150`  `200`
##   <chr>    <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 sub12    WL      0.0833 0      0.333 0.0833 0.167 0.0833 0.417
## 2 sub5     SL      0.667 0.667 0.25  0.0833 0.167 0.167 0
## 3 sub11    TOGGLE  1      1.18 1.45  1.75  1.58  2.12  1.9
## 4 sub13    SL      0.5    0.75 0.333 0.0833 0.25  0.167 0.0833
## 5 sub16    WL      0      0     0.25 0.0833 0.167 0.0833 0.417
## 6 sub4     SL      0.417 0.5   0.167 0      0.167 0.0909 0
## 7 sub7     WL      0      0     0.25 0.0833 0.167 0.0833 0.417
## 8 sub8     TOGGLE  0      0     0.25 0      0      0.0833 0.0833
## 9 sub9     SL      0.833 0.917 0.583 0.0833 0.167 0.25  0.0833
## 10 sub6    TOGGLE  0      0     0     0      0.0833 0.0833 0.0833
## 11 sub10   WL      0      0     0.333 0.0833 0.167 0.167 0.417
## 12 sub14   SL      0.583 0.75 0.333 0.167 0.0833 0.25  0
## 13 sub1    SL      0.833 0.917 0.5   0.167 0.0833 0.417 0.0833
```

```
plot(colMeans(abs_err_means[, 3:9]), ylim = c(0,.6), type = "o", xlab = "Distance (increasing)", ylab = "Abs error in number estimate")
##just doing std error for quick look
ses <- apply(abs_err_means[, 3:9], 2, std.error)
arrows(x0 = 1:7, y0 = colMeans(abs_err_means[, 3:9]) - ses, y1 = colMeans(abs_err_means[, 3:9]) + ses,
       code = 3, angle = 90, length = .1)
```



## Look at just WL condition

```
filtered_abs_err_means <- abs_err_means %>%
  filter(condition == "WL")
print(filtered_abs_err_means)

## # A tibble: 4 x 9
##   subject condition  `15`  `25`  `50`  `75`  `100`  `150`  `200`
##   <chr>    <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 sub12    WL      0.0833  0 0.333 0.0833 0.167 0.0833 0.417
## 2 sub16    WL      0      0 0.25 0.0833 0.167 0.0833 0.417
## 3 sub7     WL      0      0 0.25 0.0833 0.167 0.0833 0.417
## 4 sub10    WL      0      0 0.333 0.0833 0.167 0.167 0.417

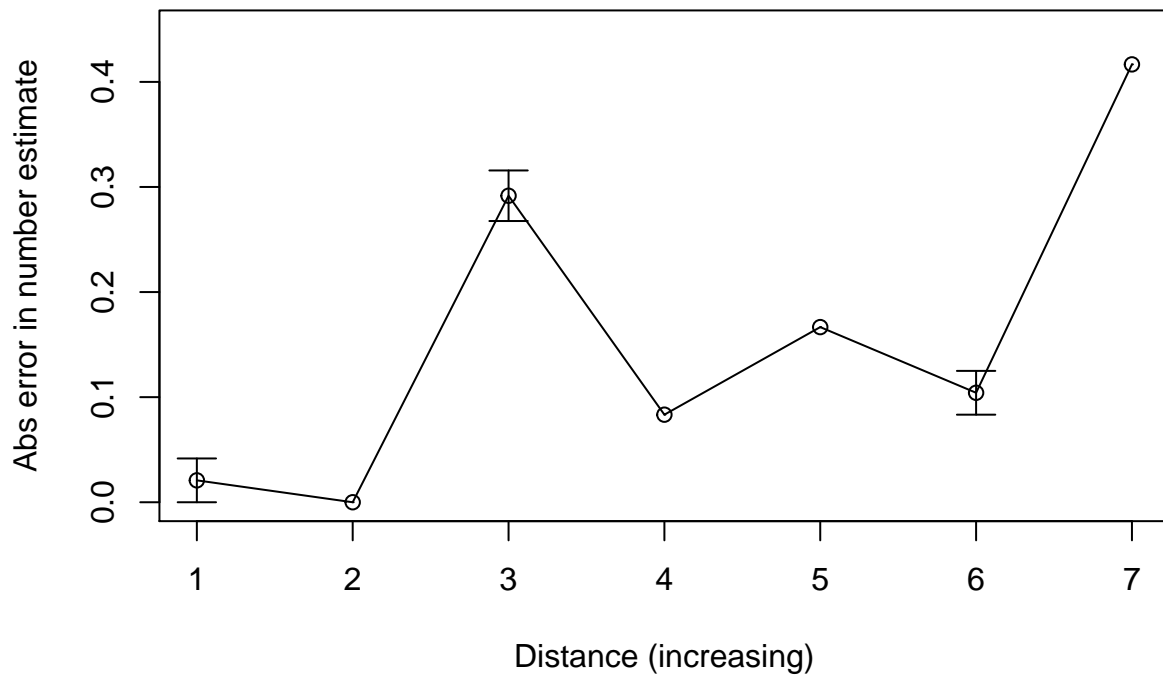
plot(colMeans(filtered_abs_err_means[ , 3:9]), ylim = c(0,.45), type = "o", xlab = "Distance (increasing)",
     ##just doing std error for quick look
     ses <- apply(filtered_abs_err_means[ , 3:9], 2, std.error)
     arrows(x0 = 1:7, y0 = colMeans(filtered_abs_err_means[ , 3:9]) - ses, y1 = colMeans(filtered_abs_err_means[ , 3:9]) + ses,
           code = 3, angle = 90, length = .1))

## Warning in arrows(x0 = 1:7, y0 = colMeans(filtered_abs_err_means[, 3:9]) - :
## zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(x0 = 1:7, y0 = colMeans(filtered_abs_err_means[, 3:9]) - :
## zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(x0 = 1:7, y0 = colMeans(filtered_abs_err_means[, 3:9]) - :
## zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(x0 = 1:7, y0 = colMeans(filtered_abs_err_means[, 3:9]) - :
## zero-length arrow is of indeterminate angle and so skipped
```



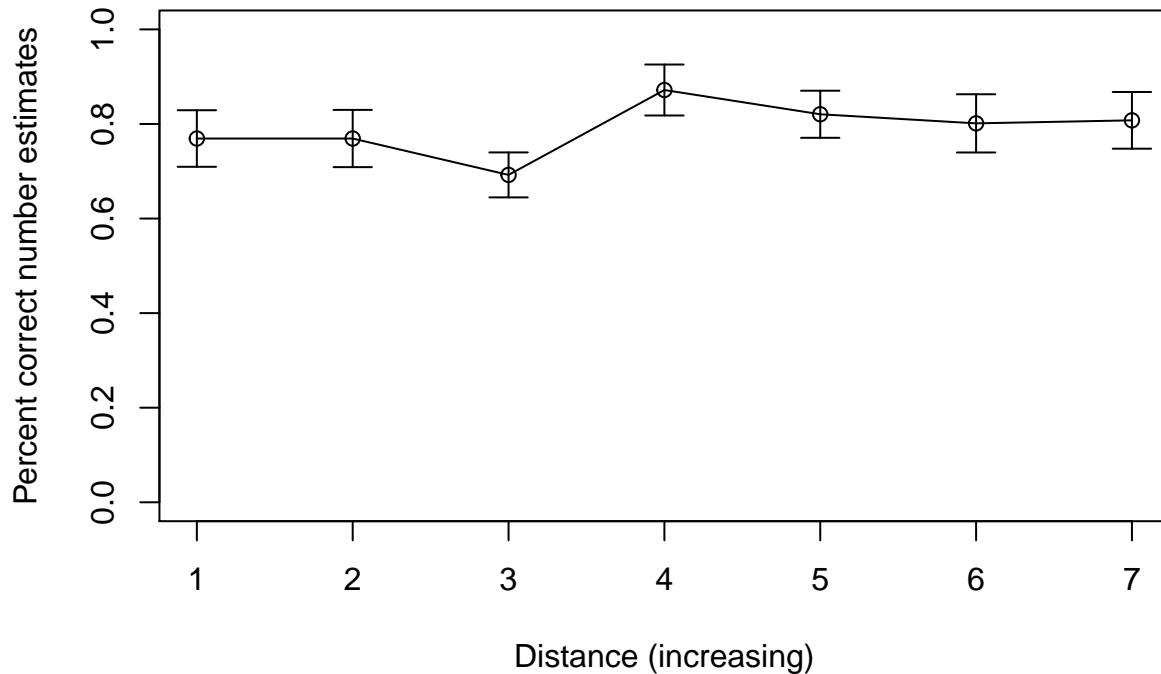
```
## Look at % correct instead
```

```
filtered_pcts <- numdat %>%
  filter(subject != "sub15" & subject != "sub2" & subject != "sub3")

pct_means <- filtered_pcts %>%
  select(subject, condition, stim_distance, iscorrect) %>%
  pivot_wider(names_from = stim_distance, values_from = iscorrect, values_fn = mean, id_cols = c(subject, condition))
dplyr::select(pct_means, `15`, `25`, `50`, `75`, `100`, `150`, `200`)
print(pct_means)
```

```
## # A tibble: 13 x 9
##   subject condition `15` `25` `50` `75` `100` `150` `200`
##   <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 sub12 WL 0.917 1 0.667 0.917 0.833 0.917 0.667
## 2 sub5 SL 0.583 0.583 0.75 0.917 0.833 0.833 1
## 3 sub11 TOGGLE 0.583 0.417 0.25 0.25 0.25 0.0833 0.25
## 4 sub13 SL 0.667 0.667 0.667 0.917 0.833 0.833 0.917
## 5 sub16 WL 1 0.917 0.75 0.917 0.833 0.917 0.667
## 6 sub4 SL 0.667 0.667 0.833 1 0.833 0.833 1
## 7 sub7 WL 1 1 0.75 0.917 0.833 0.917 0.667
## 8 sub8 TOGGLE 1 1 0.75 1 1 0.917 0.917
## 9 sub9 SL 0.583 0.583 0.583 0.917 0.833 0.833 0.917
## 10 sub6 TOGGLE 1 1 1 1 0.917 0.917 0.917
## 11 sub10 WL 1 1 0.667 0.917 0.833 0.833 0.667
## 12 sub14 SL 0.5 0.583 0.75 0.833 0.917 0.833 1
## 13 sub1 SL 0.5 0.583 0.583 0.833 0.917 0.75 0.917
```

```
plot(colMeans(pct_means[ , 3:9]), ylim = c(0,1), type = "o", xlab = "Distance (increasing)", ylab = "Percent correct number estimates")
##just doing std error for quick look
ses <- apply(pct_means[ , 3:9], 2, std.error)
arrows(x0 = 1:7, y0 = colMeans(pct_means[ , 3:9]) - ses, y1 = colMeans(pct_means[ , 3:9]) + ses,
       code = 3, angle = 90, length = .1)
```



```
SL_pct_means <- pct_means %>%
  filter(condition == "SL")
print(SL_pct_means)
```

```
## # A tibble: 6 x 9
##   subject condition `15` `25` `50` `75` `100` `150` `200`
##   <chr>   <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 sub5    SL       0.583 0.583 0.75  0.917 0.833 0.833 1
## 2 sub13   SL       0.667 0.667 0.667 0.917 0.833 0.833 0.917
## 3 sub4    SL       0.667 0.667 0.833 1      0.833 0.833 1
## 4 sub9    SL       0.583 0.583 0.583 0.917 0.833 0.833 0.917
## 5 sub14   SL       0.5   0.583 0.75  0.833 0.917 0.833 1
## 6 sub1    SL       0.5   0.583 0.583 0.833 0.917 0.75  0.917
```

```
plot(colMeans(SL_pct_means[ , 3:9]), ylim = c(0,1), type = "o", xlab = "Distance (increasing)", ylab = "Percent correct number estimates")
##just doing std error for quick look
ses <- apply(SL_pct_means[ , 3:9], 2, std.error)
arrows(x0 = 1:7, y0 = colMeans(SL_pct_means[ , 3:9]) - ses, y1 = colMeans(SL_pct_means[ , 3:9]) + ses,
       code = 3, angle = 90, length = .1, col = "gray")
```

```
WL_pct_means <- pct_means %>%
```

```

filter(condition == "WL")
print(WL_pct_means)

```

```

## # A tibble: 4 x 9
##   subject condition `15` `25` `50` `75` `100` `150` `200`
##   <chr>    <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 sub12    WL        0.917 1      0.667 0.917 0.833 0.917 0.667
## 2 sub16    WL         1      0.917 0.75  0.917 0.833 0.917 0.667
## 3 sub7     WL         1      1      0.75  0.917 0.833 0.917 0.667
## 4 sub10    WL         1      1      0.667 0.917 0.833 0.833 0.667

```

```

lines(colMeans(WL_pct_means[ , 3:9]), ylim = c(0,1), type = "o", xlab = "Distance (increasing)", ylab =
##just doing std error for quick look
ses <- apply(WL_pct_means[ , 3:9], 2, std.error)
arrows(x0 = 1:7, y0 = colMeans(WL_pct_means[ , 3:9]) - ses, y1 = colMeans(WL_pct_means[ , 3:9]) + ses,
       code = 3, angle = 90, length = .1)

```

```

## Warning in arrows(x0 = 1:7, y0 = colMeans(WL_pct_means[, 3:9]) - ses, y1 =
## colMeans(WL_pct_means[, : zero-length arrow is of indeterminate angle and so
## skipped

```

```

## Warning in arrows(x0 = 1:7, y0 = colMeans(WL_pct_means[, 3:9]) - ses, y1 =
## colMeans(WL_pct_means[, : zero-length arrow is of indeterminate angle and so
## skipped

```

```

## Warning in arrows(x0 = 1:7, y0 = colMeans(WL_pct_means[, 3:9]) - ses, y1 =
## colMeans(WL_pct_means[, : zero-length arrow is of indeterminate angle and so
## skipped

```



