RecursionAnalysis

Junyi Chu, Rose M. Schneider, Pierina Cheung

10/11/2018

Contents

Setup

R Version for citation is: `R.version.string`.

Loading data

```
#original data
full.data <- read.csv('data/recursion_full.csv', na.strings=c(""," ","NA", "NA "))
```

Reasons for exclusion and their numbers

```
full.data %>%
  dplyr::filter(ExclusionGroup != "include") %>%
  dplyr::distinct(LadlabID, .keep_all = TRUE) %>%
  dplyr::group_by(ExclusionGroup) %>%
  dplyr::summarize(countN = dplyr::n_distinct(LadlabID)) %>%
  kable()
```

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

ExclusionGroup	countN
age	5
dnf infinity	4
dnf WCN	6
experimenter error	3
L1 not english	4
parent interference	1
pilot order	3

RMS original code for checking who failed practice - do not need to run. EVAL set to FALSE

```
# check how many failed both practice trials
x <- full.data %>%
  filter(Task == "WCN" &
           (TaskItem == 1 | TaskItem == 5))%>%
  group_by(LadlabID)%>%
  summarise(sum = sum(Accuracy))%>%
  filter(sum != 2)

#just hardcoding kids because it's easier than going back to the full data frame
#These kids got 1 right, 5 wrong:
one.corr <- as.vector(c("012316-BO", "022616-JM", "030216-ED",
                        "030817-ZI", "031516-A", "032216-JH",
                        "032216-RC", "040317-AL", "040317-SL",
                        "041316-AR", "041316-NC", "041316-VN",
                        "062416-MC"))

five.corr <- as.vector("050617-Z1")

zero.corr <- as.vector(c("030216-AD", "040616-K"))
```

Exclude those who failed the practice trials on What Comes Next Task

```
#final decision: remove kids who fail 1 trial out of 2 trials, with no additional information about whe

# added 022516-ML on 2018-09-03 to the exclusion list. kid had NA data for wcn practice trials and 0 co

full.data %<>%
  mutate(ExclusionGroup = ifelse(LadlabID == "022616-JM"| LadlabID == "030216-AD"|
                                   LadlabID == "031516-A" | LadlabID == "041316-AR"|
                                   LadlabID == "041316-VN" | LadlabID == "032216-RC"|
                                   LadlabID == "012316-BO"| LadlabID == "041316-NC"|
                                   LadlabID == "022516-ML", "fail wcn", levels(ExclusionGroup)[Exclusio
         ExclusionGroup = as.factor(ExclusionGroup))

# check. good
# full.data %>%
#   filter(LadlabID == "022616-JM")
```

Let's remove anyone who should not be included in the final dataset.

```
full.data %<>%
  dplyr::filter(ExclusionGroup == "include")
```

Now, add in the Productivity classification, IHC, and FHC from PC, JC, and RMS coding

Question from RMS: is FHC capped at 100 as well? Yes, capping it at 100 for now - PC. Junyi - thoughts?

```
#productivity, fhc, ihc coding from pc, jc, and rms
hc.data <- read.csv('data/HC-datawide-forcoding - hc.datawide.csv') %>%
  dplyr::select(LadlabID, prod_tomerge, ihc_tomerge, fhc_tomerge, dce)

full.data <- dplyr::left_join(full.data, hc.data, by = "LadlabID")
```

```
## Warning: Column `LadlabID` joining factors with different levels, coercing
## to character vector
```

```r
full.data %<>%
  dplyr::rename(Productivity = prod_tomerge,
                IHC = ihc_tomerge,
                FHC = fhc_tomerge,
                DCE = dce)%>%
  dplyr::mutate(Productivity = factor(Productivity, levels = c("nonprod", "prod"),
                                      labels = c("Nonproductive", "Productive")),
                IHC = ifelse(IHC > 100, 100, IHC),
                FHC = ifelse(FHC > 100, 100, FHC))
```

Number of kids by age group and average age

```r
full.data %>%
  dplyr::group_by(AgeGroup) %>%
  dplyr::summarize(sumAge = n_distinct(LadlabID)) %>%
  kable()
```

AgeGroup	sumAge
4-4.5y	32
4.5-5y	29
5-5.5y	32
5.5-6y	29

```r
full.data %>%
  dplyr::distinct(LadlabID, .keep_all = TRUE) %>%
  dplyr::summarize(minAge = min(Age),
                   maxAge = max(Age),
                   meanAge = mean(Age),
                   sdAge = sd(Age)) %>%
  kable()
```

minAge	maxAge	meanAge	sdAge
4	5.99	4.998115	0.5713336

Number of kids who were classified as decade productive & nonproductive

```r
full.data %>%
  dplyr::distinct(LadlabID, Productivity, Age)%>%
  dplyr::group_by(Productivity)%>%
  dplyr::summarise(n = n(),
                   meanage = mean(Age, na.rm=TRUE),
                   sdage = sd(Age, na.rm=TRUE),
                   minage = min(Age, na.rm=TRUE),
                   maxage = max(Age, na.rm=TRUE)) %>%
  kable()
```

Productivity	n	meanage	sdage	minage	maxage
Nonproductive	43	4.598837	0.4192125	4.00	5.61
Productive	79	5.215443	0.5253760	4.05	5.99

Just for reference, this is the number of kids who switched classifications from PC, JC, RMS recode

```
full.data %>%
  dplyr::filter(TaskType == "productivity") %>%
  droplevels()%>%
  dplyr::distinct(LadlabID, Response, Productivity) %>%
  dplyr::mutate(Response = factor(Response, levels = c("nonprod", "prod"),
                                  labels = c("Nonproductive", "Productive")))%>%
  dplyr::mutate(changed_classification = ifelse((is.na(Response) & Productivity == "Nonproductive"),
                                                "NA_toNonprod",
                                                ifelse((is.na(Response) & Productivity == "Productive")
                                                       "NA_toProd",
                                                       ifelse((Response == "Nonproductive" & Productivi
                                                              "Nonprod_toProd",
                                                              ifelse((Response == "Productive" & Produc
                                                                     "Prod_toNonprod", "no_change")))))
  dplyr::group_by(changed_classification)%>%
  dplyr::summarise(n =n())
```

```
## # A tibble: 4 x 2
##   changed_classification     n
##   <chr>                  <int>
## 1 NA_toNonprod              29
## 2 NA_toProd                  7
## 3 no_change                 85
## 4 Nonprod_toProd             1
```

Highest Count Descriptives

Average of IHC, DCE, and FHC for all kids

```
full.data %>%
  dplyr::distinct(LadlabID, IHC)%>%
  dplyr::summarise(mean_IHC = mean(IHC),
                   sd_IHC = sd(IHC),
                   min_IHC = min(IHC),
                   max_IHC = max(IHC),
                   median_IHC = median(IHC)) %>%
  kable()
```

mean_IHC	sd_IHC	min_IHC	max_IHC	median_IHC
50.41803	33.80568	5	100	39.5

```
full.data %>%
  dplyr::distinct(LadlabID, DCE)%>%
  dplyr::summarise(mean_DCE = mean(DCE, na.rm=TRUE),
                   sd_DCE = sd(DCE, na.rm=TRUE),
                   min_DCE = min(DCE, na.rm=TRUE),
                   max_DCE = max(DCE, na.rm=TRUE),
                   median_DCE = median(DCE, na.rm=TRUE)) %>%
  kable()
```

mean_DCE	sd_DCE	min_DCE	max_DCE	median_DCE
43.80769	17.54438	19	99	44

```
full.data %>%
  dplyr::distinct(LadlabID, FHC)%>%
  dplyr::summarise(mean_FHC = mean(FHC),
                   sd_FHC = sd(FHC),
                   min_FHC = min(FHC),
                   max_FHC = max(FHC),
                   median_FHC = median(FHC)) %>%
  kable()
```

mean_FHC	sd_FHC	min_FHC	max_FHC	median_FHC
71.55738	34.64532	5	100	99

Similar data by decade productivity

```
full.data %>%
  dplyr::distinct(LadlabID, Productivity, IHC)%>%
  dplyr::group_by(Productivity)%>%
  dplyr::summarise(mean_IHC = mean(IHC),
                   sd_IHC = sd(IHC),
                   min_IHC = min(IHC),
                   max_IHC = max(IHC),
                   median_IHC = median(IHC)) %>%
  kable()
```

Productivity	mean_IHC	sd_IHC	min_IHC	max_IHC	median_IHC
Nonproductive	23.76744	15.20156	5	77	18
Productive	64.92405	32.30693	12	100	59

```
full.data %>%
  dplyr::distinct(LadlabID, Productivity, DCE)%>%
  dplyr::group_by(Productivity)%>%
  dplyr::summarise(mean_DCE = mean(DCE, na.rm=TRUE),
                   sd_DCE = sd(DCE, na.rm=TRUE),
                   min_DCE = min(DCE, na.rm=TRUE),
                   max_DCE = max(DCE, na.rm=TRUE),
                   median_DCE = median(DCE, na.rm=TRUE)) %>%
  kable()
```

Productivity	mean_DCE	sd_DCE	min_DCE	max_DCE	median_DCE
Nonproductive	29.62500	8.539126	19	49	29
Productive	50.11111	16.865481	19	99	49

```
full.data %>%
  dplyr::distinct(LadlabID, Productivity, FHC)%>%
  dplyr::group_by(Productivity)%>%
```

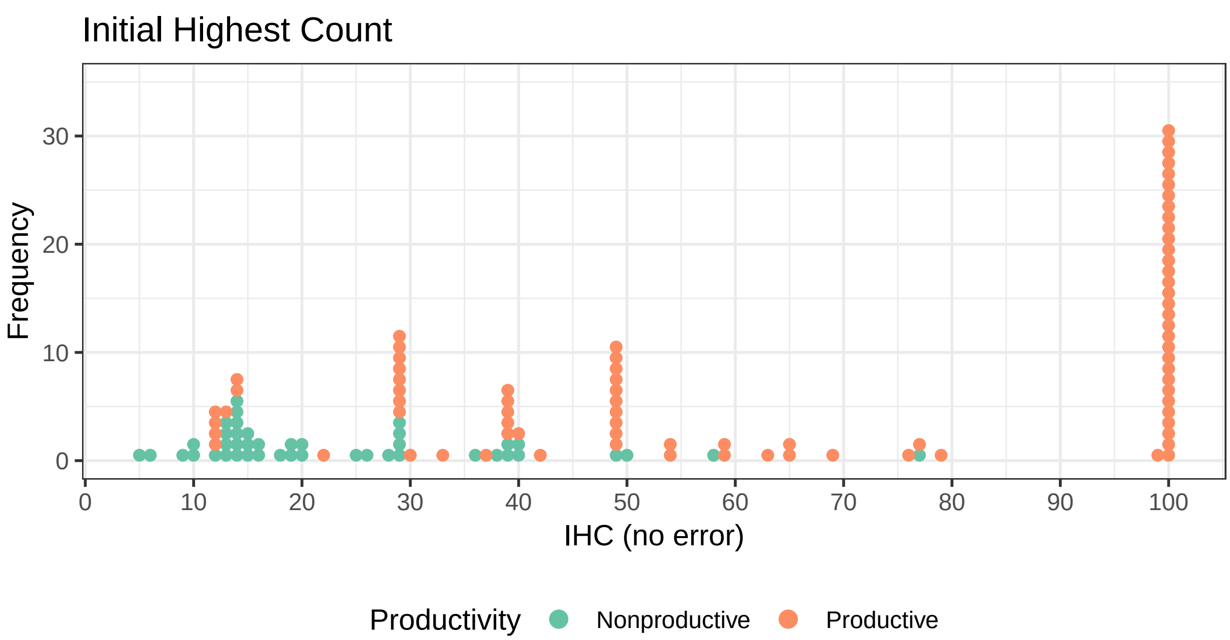
```
dplyr::summarise(mean_FHC = mean(FHC),
                 sd_FHC = sd(FHC),
                 min_FHC = min(FHC),
                 max_FHC = max(FHC),
                 median_FHC = median(FHC)) %>%
  kable()
```

Productivity	mean_FHC	sd_FHC	min_FHC	max_FHC	median_FHC
Nonproductive	29.13953	14.46438	5	77	29
Productive	94.64557	14.74922	38	100	100

Plotting distribution of IHC, as a function of productivity (~ junyi's graph)

```
unique.hc.data <- full.data %>%
  dplyr::distinct(LadlabID, Gender, Age, AgeGroup, HCReceivedSupport, IHC, DCE, FHC, Productivity)

ggplot(unique.hc.data, aes(x=IHC, color=Productivity)) +
  geom_dotplot(aes(fill = Productivity),
               binwidth=1, stackgroups=TRUE, binpositions="all",method="dotdensity") +
  scale_color_brewer(palette="Set2") +
  scale_fill_brewer(palette="Set2") +
  coord_fixed(ratio=1) +
  scale_y_continuous(breaks=seq(0,40,10), lim=c(0,35)) +
  scale_x_continuous(breaks=seq(0,100,by=10)) +
  labs(title="Initial Highest Count",
       x="IHC (no error)",
       y="Frequency") +
  theme_bw() +
  theme(legend.position="bottom")
```




```
ggsave('graphs/ihc-by-prod.png')
```

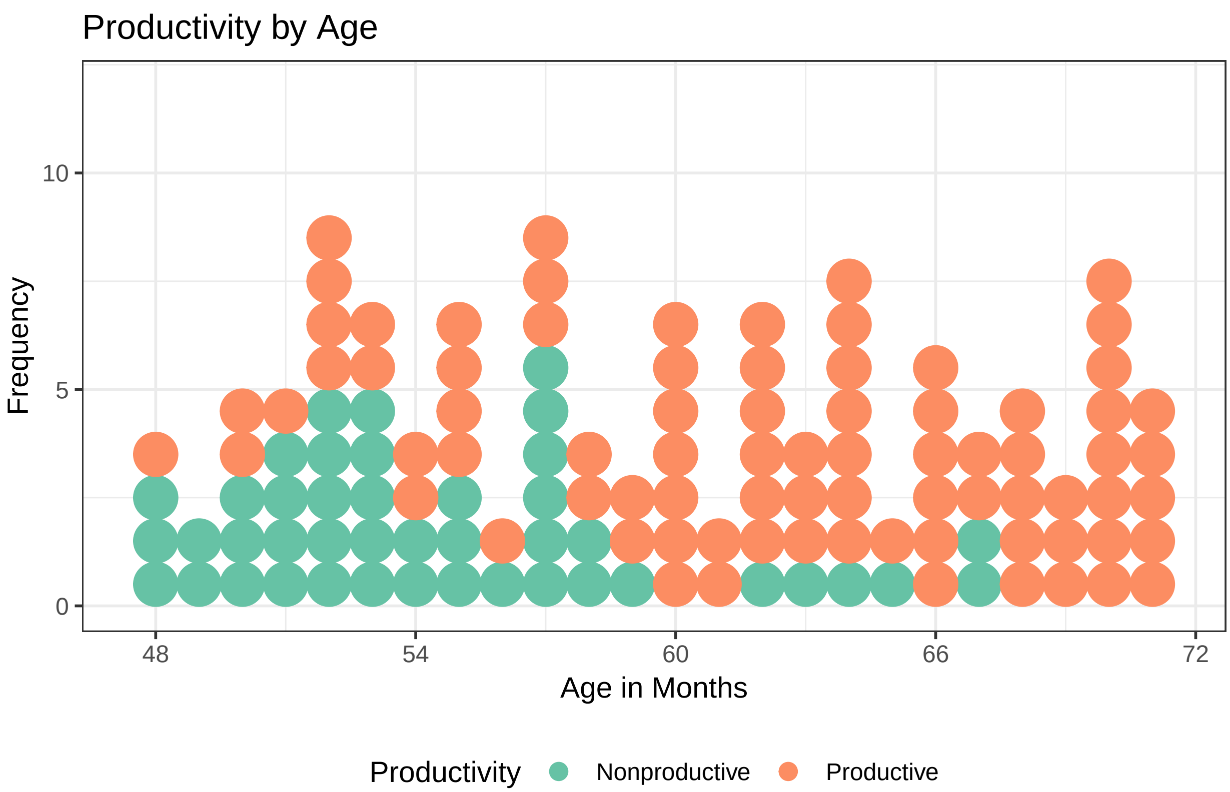
```
## Saving 6.5 x 4.5 in image
```

```r
# hist(unique.hc.data$IHC)
```

Plotting productivity as a function of age in months

```r
unique.hc.data$AgeMonths = floor(unique.hc.data$Age*12)

ggplot(unique.hc.data, aes(x=AgeMonths, colour = Productivity)) +
  geom_dotplot(aes(fill = Productivity),
               binwidth = 1,
               stackgroups=TRUE, binpositions="all") +
  coord_fixed(ratio=1) +
  scale_y_continuous(breaks=seq(0,10,5), lim=c(0,12)) +
  scale_x_continuous(breaks=seq(48,72,by=6)) +
  scale_color_brewer(palette="Set2") +
  scale_fill_brewer(palette="Set2") +
  labs(title="Productivity by Age",
       x="Age in Months",
       y="Frequency") +
  theme_bw() +
  theme(legend.position="bottom")
```




```r
ggsave('graphs/prod-by-age.png')
```

```
## Saving 6.5 x 4.5 in image
```

Distance between IHC and FHC

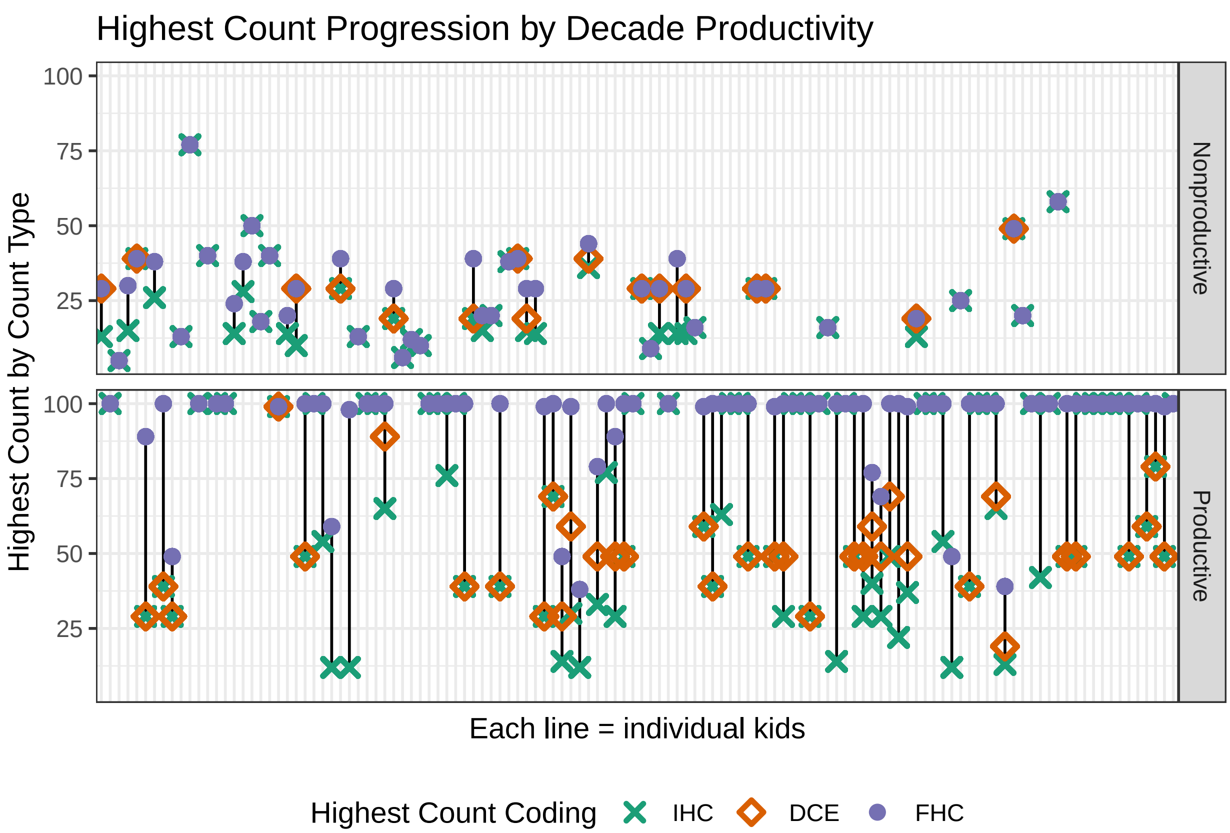
Restruture data to plot distance between IHC, DCE, and FHC

```r
hc.dev.data <- full.data %>%
  dplyr::select(LadlabID, Age, Productivity, IHC, DCE, FHC) %>%
```

```
  gather(hcprogression, hc, IHC:FHC)%>%
  mutate(hcprogression = factor(hcprogression, levels = c("IHC", "DCE", "FHC"))) %>%
  rename(`Highest Count Coding` = hcprogression)

#all kids together
ggplot(hc.dev.data, aes(x = LadlabID, y = hc)) +
  facet_grid(rows = vars(Productivity)) +
  geom_line(data=hc.dev.data[!is.na(hc.dev.data$hc),]) +
  geom_point(aes(shape = `Highest Count Coding`, colour = `Highest Count Coding`),
             size = 2, stroke = 1.5) +
  scale_color_brewer(palette="Dark2") +
  scale_shape_manual(values = c(4,5,20)) +
  labs(title="Highest Count Progression by Decade Productivity",
       x = "Each line = individual kids",
       y="Highest Count by Count Type") +
  theme_bw() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 270, hjust = 1)) +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

```
## Warning: Removed 3010 rows containing missing values (geom_point).
```

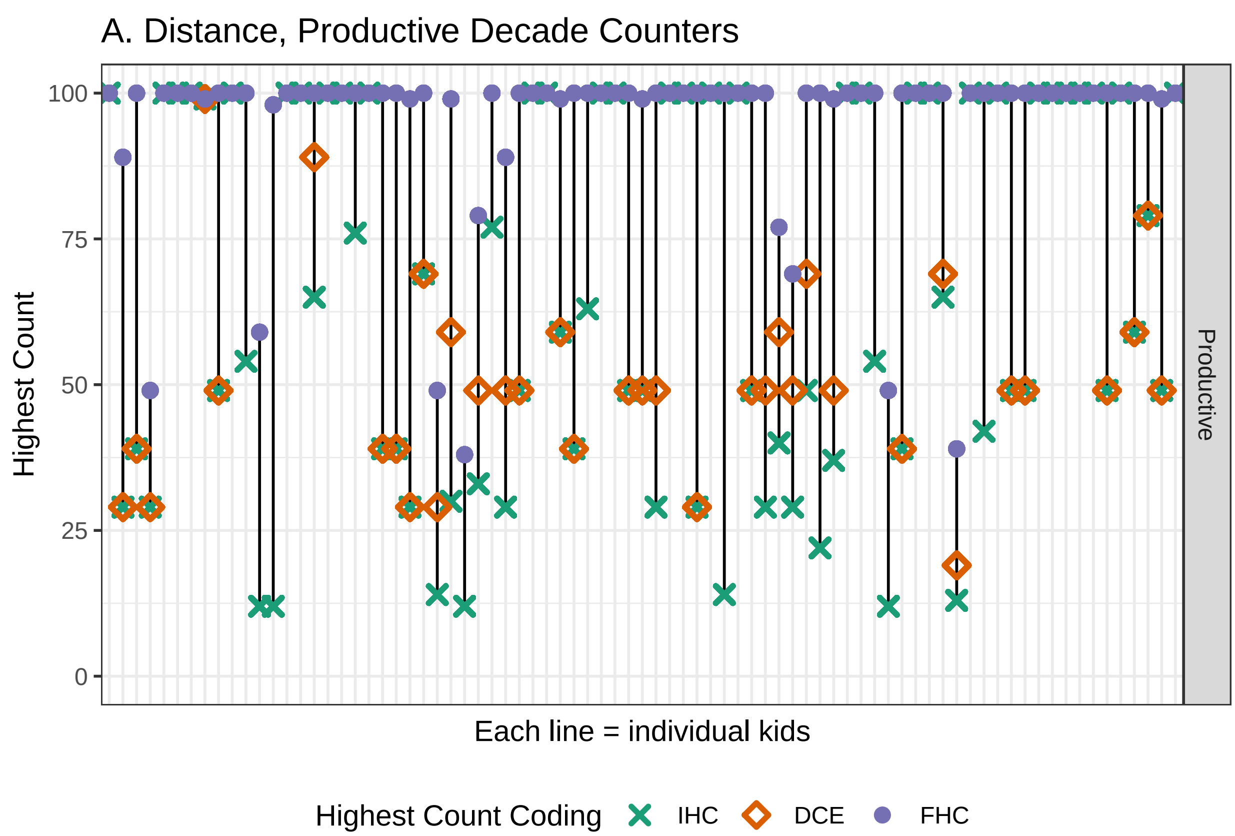



```
hc.dev.prod <- subset(hc.dev.data, Productivity == "Productive")
hc.dev.nonprod <- subset(hc.dev.data, Productivity == "Nonproductive")
```

Separate graphs for productivity groups (for easier viewing)

```r
#productive
ggplot(hc.dev.prod, aes(x = LadlabID, y = hc)) +
  facet_grid(rows = vars(Productivity)) +
  geom_line(data=hc.dev.prod[!is.na(hc.dev.prod$hc),]) +
  geom_point(aes(shape = `Highest Count Coding`, colour = `Highest Count Coding`),
             size = 2, stroke = 1.5) +
  scale_color_brewer(palette="Dark2") +
  scale_shape_manual(values = c(4,5,20)) +
  ylim(0, 100) +
  labs(title="A. Distance, Productive Decade Counters",
       x = "Each line = individual kids",
       y="Highest Count") +
  theme_bw() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 270, hjust = 1)) +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

```
## Warning: Removed 1849 rows containing missing values (geom_point).
```



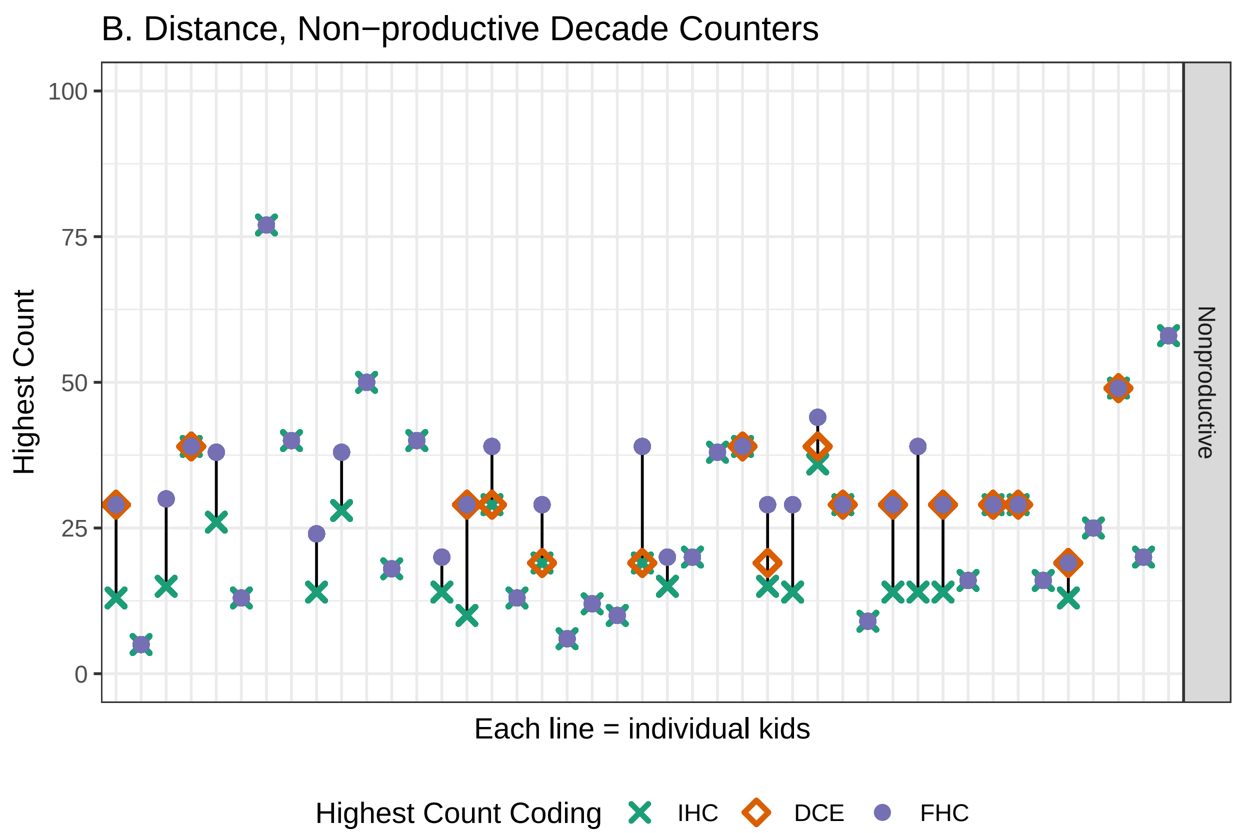

```r
ggsave('graphs/distance-prod.png')
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 1849 rows containing missing values (geom_point).
```

```r
#nonproductive
ggplot(hc.dev.nonprod, aes(x = LadlabID, y = hc)) +
  facet_grid(rows = vars(Productivity)) +
```

```
  geom_line(data=hc.dev.nonprod[!is.na(hc.dev.nonprod$hc),]) +
  geom_point(aes(shape = `Highest Count Coding`, colour = `Highest Count Coding`),
             size = 2, stroke = 1.5) +
  scale_color_brewer(palette="Dark2") +
  scale_shape_manual(values = c(4,5,20)) +
  ylim(0, 100) +
  labs(title="B. Distance, Non-productive Decade Counters",
       x = "Each line = individual kids",
       y="Highest Count") +
  theme_bw() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 270, hjust = 1)) +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

```
## Warning: Removed 1161 rows containing missing values (geom_point).
```




```
ggsave('graphs/distance-nonprod.png')
```

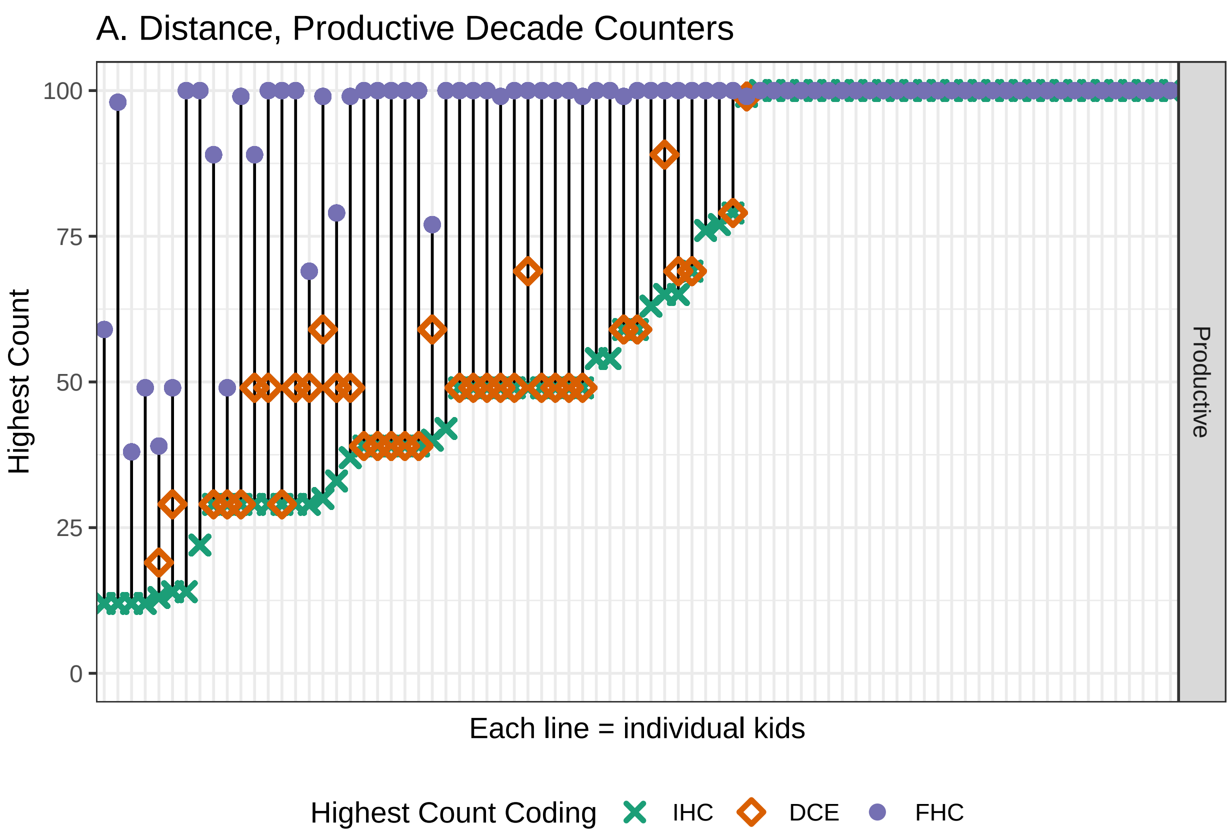
```
## Saving 6.5 x 4.5 in image
## Warning: Removed 1161 rows containing missing values (geom_point).
```

Separate graphs for productivity groups, sorted by ascending IHC

```
#productive
ggplot(hc.dev.prod, aes(x = reorder(LadlabID, hc, FUN=min), y = hc)) +
  facet_grid(rows = vars(Productivity)) +
  geom_line(data=hc.dev.prod[!is.na(hc.dev.prod$hc),]) +
```

```r
  geom_point(aes(shape = `Highest Count Coding`, colour = `Highest Count Coding`),
             size = 2, stroke = 1.5) +
  scale_color_brewer(palette="Dark2") +
  scale_shape_manual(values = c(4,5,20)) +
  ylim(0, 100) +
  labs(title="A. Distance, Productive Decade Counters",
       x = "Each line = individual kids",
       y="Highest Count") +
  theme_bw() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 270, hjust = 1)) +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

```
## Warning: Removed 1849 rows containing missing values (geom_point).
```



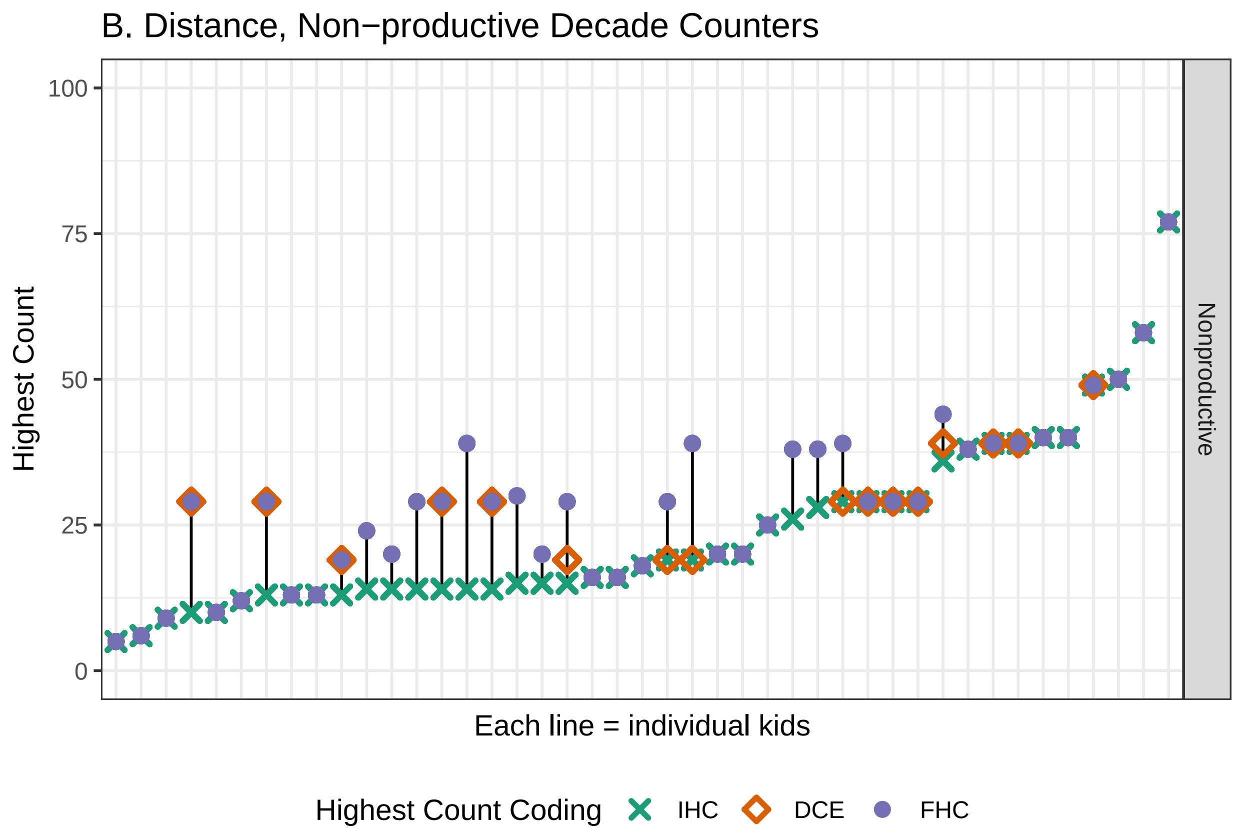

```r
ggsave('graphs/distance-prod-sorted.png')
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 1849 rows containing missing values (geom_point).
```

```r
#nonproductive
ggplot(hc.dev.nonprod, aes(x = reorder(LadlabID, hc, FUN=min), y = hc)) +
  facet_grid(rows = vars(Productivity)) +
  geom_line(data=hc.dev.nonprod[!is.na(hc.dev.nonprod$hc),]) +
  geom_point(aes(shape = `Highest Count Coding`, colour = `Highest Count Coding`),
             size = 2, stroke = 1.5) +
  scale_color_brewer(palette="Dark2") +
```

```
  scale_shape_manual(values = c(4,5,20)) +
  ylim(0, 100) +
  labs(title="B. Distance, Non-productive Decade Counters",
       x = "Each line = individual kids",
       y="Highest Count") +
  theme_bw() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 270, hjust = 1)) +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

```
## Warning: Removed 1161 rows containing missing values (geom_point).
```




```
ggsave('graphs/distance-nonprod-sorted.png')
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 1161 rows containing missing values (geom_point).
```

Number of kids who counted to 99+ spontaneously on IHC plus those whose FHC = 99+ without prompting

```
# full.data %>%
#   filter(IHC > 98) %>%
#   distinct(LadlabID, IHC, FHC, HCReceivedSupport) %>%
#   count() #n=32
# but some kids made errors past IHC but < 3 so need to account for that

full.data %>%
  filter(FHC > 98 & (is.na(HCReceivedSupport)|HCReceivedSupport != 1)) %>%
```

```r
  distinct(LadlabID, IHC, FHC, HCReceivedSupport) %>%
  count() #n =42
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    42
```

Average number of decade prompts provided. Productive counters first

```r
full.data %>%
  filter(TaskItem == "times") %>%
  filter(Productivity == "Productive") %>%
  distinct(LadlabID, HCReceivedSupport, TaskItem, Response) %>%
  mutate(Response = as.numeric(levels(Response)[Response])) %>%
  group_by(HCReceivedSupport) %>%
  summarize(mean = mean(Response, na.rm=TRUE),
            sd = sd(Response, na.rm=TRUE),
            min = min(Response, na.rm=TRUE),
            max = max(Response, na.rm=TRUE),
            count=n())
```

```
## # A tibble: 3 x 6
##   HCReceivedSupport   mean     sd   min   max count
##   <fct>              <dbl>  <dbl> <dbl> <dbl> <int>
## 1 0                   2     NA        2     2    37
## 2 1                   3.32   1.73     1     7    37
## 3 <NA>              NaN    NaN      Inf  -Inf     5
```

```r
# assume 0 = NA
# error in supported.times coding, should only count to 90
# but one kid got prompted with 100 and 110 and times should be 0
```

Then nonproductive counters.

```r
full.data %>%
  filter(TaskItem == "times") %>%
  filter(Productivity == "Nonproductive") %>%
  distinct(LadlabID, HCReceivedSupport, TaskItem, Response) %>%
  mutate(Response = as.numeric(levels(Response)[Response])) %>%
  group_by(HCReceivedSupport) %>%
  summarize(mean = mean(Response, na.rm=TRUE),
            sd = sd(Response, na.rm=TRUE),
            min = min(Response, na.rm=TRUE),
            max = max(Response, na.rm=TRUE),
            count=n())
```

```
## # A tibble: 2 x 6
##   HCReceivedSupport  mean    sd   min   max count
##   <fct>             <dbl> <dbl> <dbl> <dbl> <int>
## 1 0                  1.33 0.577     1     2    29
## 2 1                  1.29 0.469     1     2    14
```

```r
# assume 0 = NA
# error in supported.times coding, should only count to 90
# but one kid got prompted with 100 and 110 and times should be 0
```

What Comes Next Descriptives

Note minimum highest contig NN can be 5 (one of the practice trials). Practice trials are excluded from %corr and within vs. beyond computation.

First check if Accuracy column in full.data is coded correctly. Good to go.

```
wcn.data <- full.data %>%
  filter(Task == "WCN")

wcn.data %<>%
  mutate(Response_num = as.numeric(as.character(Response)),
         TaskItem_num = as.numeric(as.character(TaskItem)),
         Accuracy_check = ifelse(Response_num == (TaskItem_num + 1), 1, 0),
         Accuracy_valid = ifelse(Accuracy == Accuracy_check, TRUE, FALSE))
```

```
## Warning in evalq(as.numeric(as.character(Response)), <environment>): NAs
## introduced by coercion
```

```
validate <- function(){
  validation <- wcn.data %>%
    filter(Accuracy_valid == FALSE)
  if(length(validation$LadlabID) > 0) {
    print("WARNING: CHECK CODING")
  } else {
    print("All coding correct")
  }
}

validate()
```

```
## [1] "All coding correct"
```

Immediate vs. Momentum trials: Children were provided with momentum trials if they got wrong on immediate trials. Check %trials where immediate = wrong, momentum = right

```
wcn.wide <- full.data %>%
  filter(ExclusionGroup == "include") %>%
  filter(Task == "WCN") %>%
  filter(TaskType != "practice") %>%
  filter(TaskItem != 3) %>% # a trial on 3 for momentum that doesn't exist for immediate
  droplevels()%>%
  dplyr::select(LadlabID, Age, AgeGroup, TaskType, TaskItem, Accuracy, Productivity) %>%
  spread(TaskType, Accuracy)

# data check: some kids got 1 for immediate but 0 for momentum or 1 for immediate and 1 for momentum (N
## for reference, pulling out these kids below
full.data %>%
  filter(Task == "WCN",
         TaskType == "momentum" | TaskType == "immediate")%>%
  dplyr::select(LadlabID, Age, AgeGroup, TaskType, TaskItem, Accuracy) %>%
  spread(TaskType, Accuracy)%>%
  mutate(issue_immediate1Momentum0 = ifelse(immediate == 1 & momentum == 0, TRUE, FALSE),
         issue_immediate1Momentum1 = ifelse(immediate == 1 & momentum == 1, TRUE, FALSE))%>%
  filter(issue_immediate1Momentum0 == TRUE |
           issue_immediate1Momentum1 == TRUE)
```

```
##     LadlabID  Age AgeGroup TaskItem immediate momentum
## 1 011216-WB 4.44   4-4.5y       59         1        1
## 2 022616-AG 4.32   4-4.5y       37         1        1
## 3 031616-RP 4.84   4.5-5y       23         1        1
## 4 041316-CC 4.36   4-4.5y       62         1        0
## 5 111117-VK 5.87   5.5-6y       29         1        1
##   issue_immediate1Momentum0 issue_immediate1Momentum1
## 1                     FALSE                      TRUE
## 2                     FALSE                      TRUE
## 3                     FALSE                      TRUE
## 4                      TRUE                     FALSE
## 5                     FALSE                      TRUE
```

```
# how many kids show improved performance
xtabs(~immediate + momentum, data = wcn.wide, na.action = na.pass, exclude = NULL)
```

```
##          momentum
## immediate   0   1 <NA>
##      0    263 174   13
##      1      1   4  520
##      <NA>   1   0    0
```

```
# 191 / 1048 trials = ~ 18%. NOTE % not by kids but by trials.
```

Percent Correct on WCN

```
wcn.data %>%
  dplyr::filter(TaskType == "immediate") %>%
  dplyr::group_by(LadlabID) %>%
  dplyr::summarize(avg.wcn = mean(Accuracy, na.rm=TRUE),
                   sd.wcn = sd(Accuracy, na.rm=TRUE)) %>%
  dplyr::summarize(avg = mean(avg.wcn),
                   sd = sd(sd.wcn))
```

```
## # A tibble: 1 x 2
##     avg    sd
##   <dbl> <dbl>
## 1 0.538 0.215
```

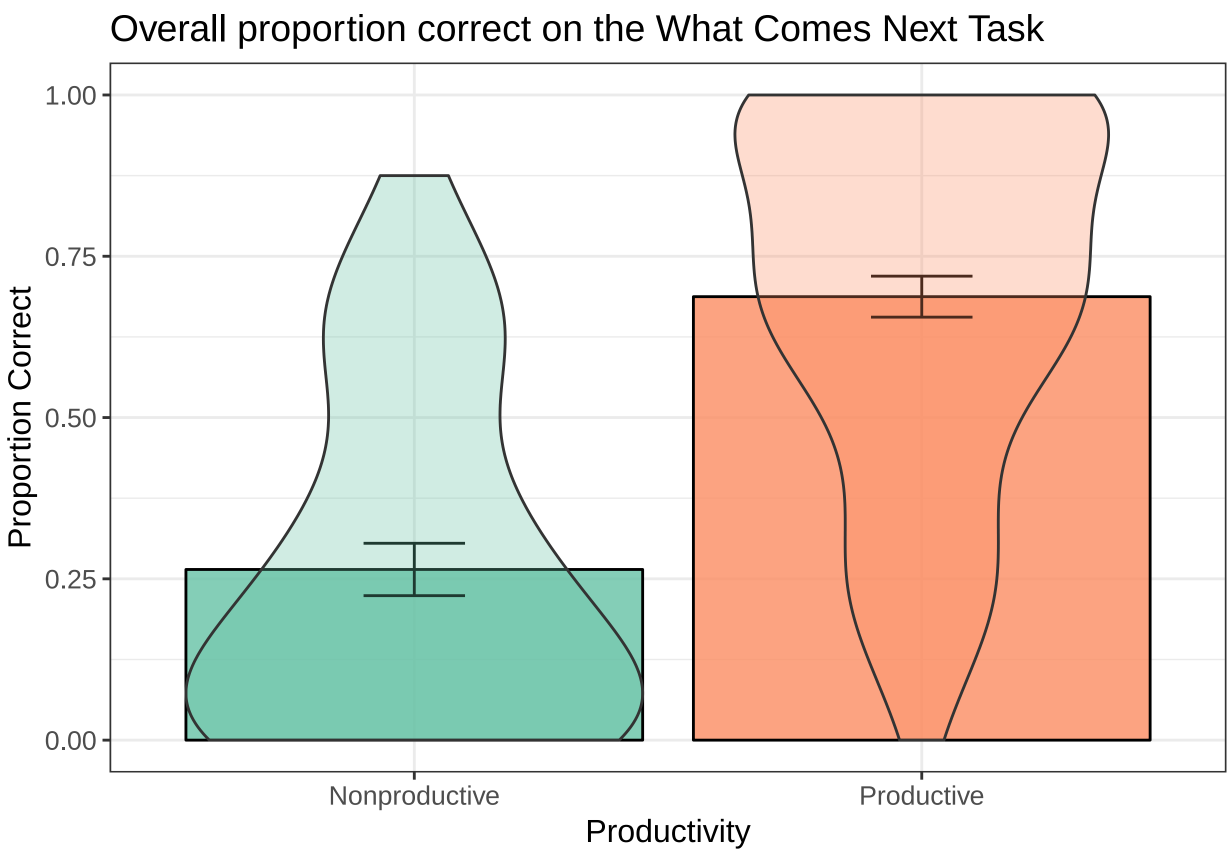
```
wcn.data %>%
  dplyr::filter(TaskType == "immediate") %>%
  dplyr::group_by(LadlabID, Productivity) %>%
  dplyr::summarize(avg.wcn = mean(Accuracy, na.rm=TRUE),
                   sd.wcn = sd(Accuracy, na.rm=TRUE)) %>%
  dplyr::group_by(Productivity) %>%
  dplyr::summarize(avg = mean(avg.wcn),
                   sd = sd(sd.wcn))
```

```
## # A tibble: 2 x 3
##   Productivity    avg    sd
##   <fct>         <dbl> <dbl>
## 1 Nonproductive 0.265 0.216
## 2 Productive    0.687 0.215
```

Plotting %corr on WCN as function of productivity

```
wcn.data %>%
  dplyr::filter(TaskType == "immediate") %>%
```

```
dplyr::group_by(LadlabID, Productivity) %>%
dplyr::summarize(avg.wcn = mean(Accuracy, na.rm=TRUE),
                 sd.wcn = sd(Accuracy, na.rm=TRUE)) %>%
ggplot(aes(x = Productivity, y = avg.wcn, fill=factor(Productivity))) +
stat_summary(fun.y = mean, position = position_dodge(width = .95),
             geom="bar", alpha = .8, colour = "black") +
stat_summary(fun.data = mean_se, geom="errorbar",
             position = position_dodge(width=0.90), width = 0.2)+
#scale_fill_discrete(name = "Productivity") +
scale_fill_brewer(name="Productivity", palette="Set2") +
ylab("Proportion Correct") +
xlab('Productivity') +
theme_bw() +
theme(legend.position = "none") +
ggtitle("Overall proportion correct on the What Comes Next Task") +
theme(text = element_text(size = 12)) +
ylim(0, 1.0) +
geom_violin(alpha = .3)
```




```
ggsave('graphs/wcn-percentcorr.png')
```

```
## Saving 6.5 x 4.5 in image
```

Add whether the Task Item was within or outside of the kid's initial highest count.

```
#first, get initial highest count for each kiddo
#Make a lookup table with SID and initial highest count
lookup <- full.data %>%
```

```r
  distinct(LadlabID, IHC)

wcn.data %<>%
  dplyr::mutate(TaskItem = as.numeric(as.character(TaskItem)))

#This is a function that, for each trial, checks the number queried. If number queried is above the chi
determine_count_range <- function(df) {
  tmp <- df
  for (row in 1:nrow(tmp)) {
    sub = as.character(tmp[row, "LadlabID"])
    count_range = as.numeric(as.character(subset(lookup, LadlabID == sub)$IHC))
    tmp[row, "IHC"] = as.numeric(as.character(count_range))
    if (tmp[row, "TaskItem"] > count_range) {
      tmp[row, "WithinOutsideIHC"] = "outside"
    } else {
      tmp[row, "WithinOutsideIHC"] = "within"
    }
  }
  return(tmp)
}

#Run for wcn
wcn.data <- determine_count_range(wcn.data)
```

WCN accuracy, within and outside of IHC

```r
wcn.data %>%
  dplyr::filter(TaskType == "immediate") %>%
  dplyr::group_by(WithinOutsideIHC) %>%
  dplyr::summarize(mean = mean(Accuracy, na.rm = TRUE),
                   sd = sd(Accuracy, na.rm = TRUE))
```

```
## # A tibble: 2 x 3
##   WithinOutsideIHC  mean    sd
##   <chr>            <dbl> <dbl>
## 1 outside          0.342 0.475
## 2 within           0.764 0.425
```

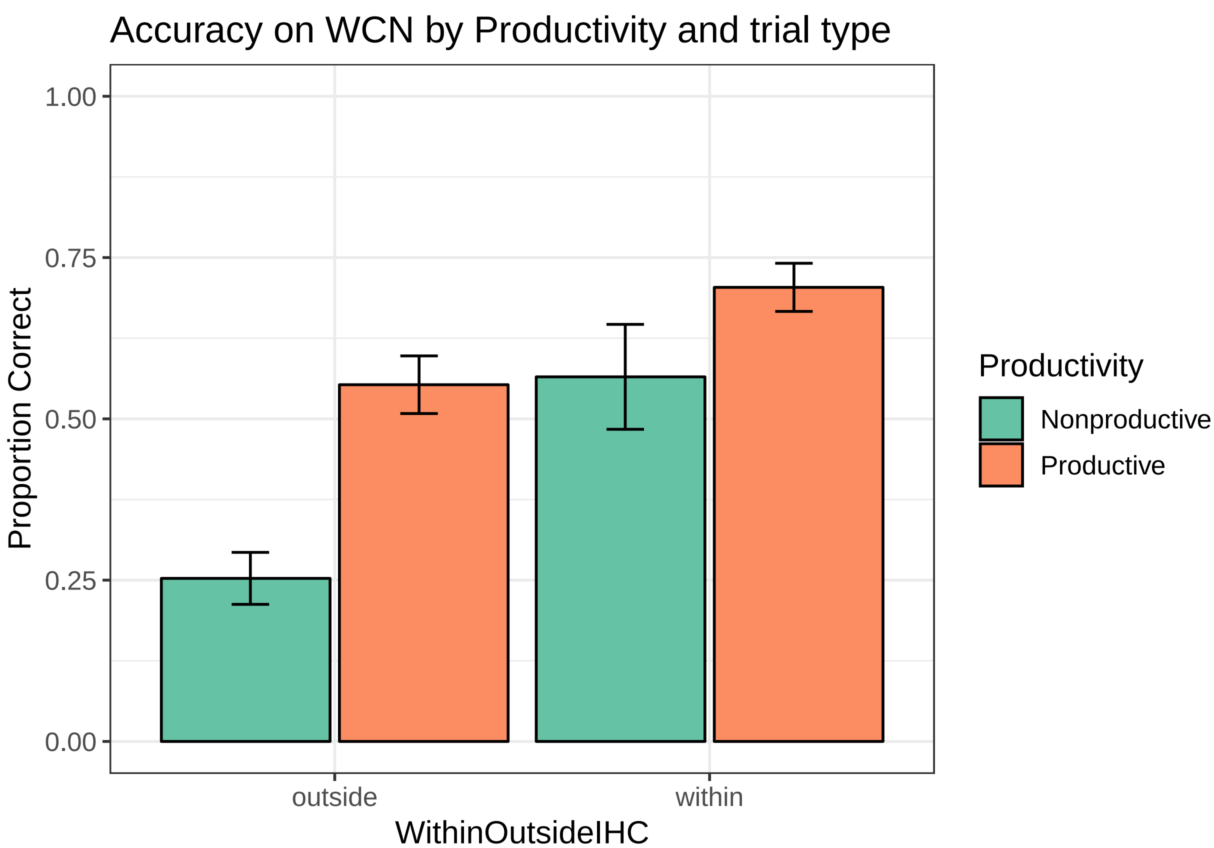
Now WCN by within/outside count range and productivity

```r
wcn.data %>%
  dplyr::filter(TaskType == "immediate") %>%
  dplyr::group_by(Productivity, WithinOutsideIHC) %>%
  dplyr::summarize(mean = mean(Accuracy, na.rm = TRUE),
                   sd = sd(Accuracy, na.rm = TRUE))
```

```
## # A tibble: 4 x 4
## # Groups:   Productivity [?]
##   Productivity  WithinOutsideIHC  mean    sd
##   <fct>         <chr>            <dbl> <dbl>
## 1 Nonproductive outside          0.207 0.406
## 2 Nonproductive within           0.612 0.492
## 3 Productive    outside          0.518 0.501
## 4 Productive    within           0.783 0.413
```

Plotting WCN as within vs. beyond by productivity

```r
wcn.data %>%
  dplyr::filter(TaskType == "immediate") %>%
  dplyr::group_by(Productivity, WithinOutsideIHC, LadlabID) %>%
  dplyr::summarize(meansubj = mean(Accuracy, na.rm = TRUE)) %>%
  ggplot(aes(x=WithinOutsideIHC, y=meansubj, fill=Productivity)) +
  stat_summary(fun.y = mean, position = position_dodge(width = .95),
               geom="bar", colour = "black") +
  stat_summary(fun.data = mean_se, geom="errorbar",
               position = position_dodge(width=0.90), width = 0.2) +
  scale_fill_brewer(name = "Productivity", palette="Set2") +
  scale_y_continuous(lim=c(0,1)) +
  labs(title="Accuracy on WCN by Productivity and trial type",
       y="Proportion Correct",
       fill="Productivity") +
  theme_bw() +
  theme(text = element_text(size = 12))
```

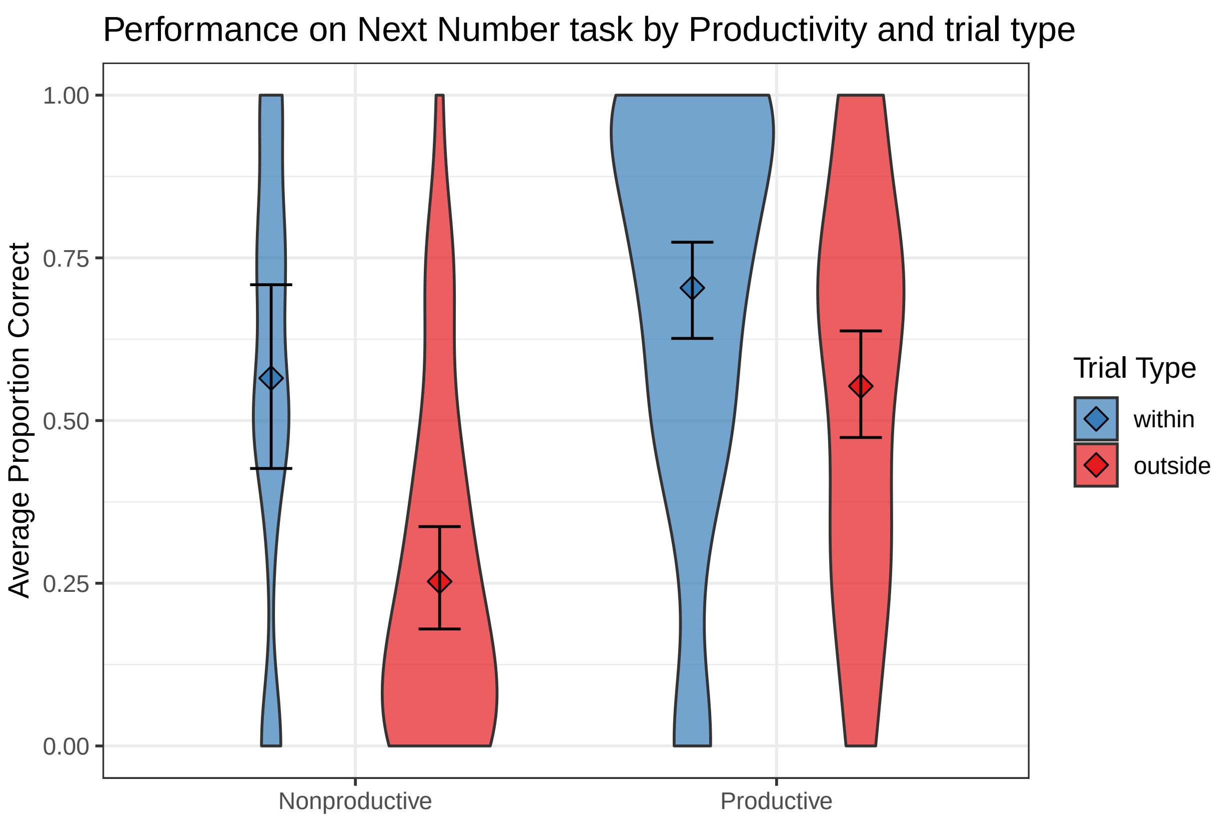



```r
ggsave('graphs/wcn-within-beyond.png')
```

```
## Saving 6.5 x 4.5 in image
```

```r
wcn.data %>%
  dplyr::filter(TaskType == "immediate") %>%
  dplyr::group_by(Productivity, WithinOutsideIHC, LadlabID) %>%
  dplyr::summarize(meansubj = mean(Accuracy, na.rm = TRUE)) %>%
  ggplot(aes(x=Productivity, y=meansubj, fill=reorder(WithinOutsideIHC, WithinOutsideIHC, length))) +
  geom_violin(alpha=0.7, scale="count", position=position_dodge(width=0.8)) +
  #geom_point(position=position_jitterdodge(jitter.width=0.2, dodge.width=.9)) +
```

```
  stat_summary(fun.y = mean,position = position_dodge(width=0.8),
               geom="point", shape=23, size=3, colour = "black") +
  stat_summary(fun.data = mean_cl_boot, geom="errorbar",
               position = position_dodge(width=0.8), width = 0.2) +
  scale_fill_brewer(name = "Trial Type", palette="Set1",direction=-1) +
  scale_y_continuous(limits=c(0,1)) +
  labs(title="Performance on Next Number task by Productivity and trial type",
       y="Average Proportion Correct", x="",
       fill="Trial Type") +
  theme_bw()
```




```
ggsave('graphs/wcn-within-beyond2.png', width=6, height=4)
```

todo

Analysis of WCN accuracy by productivity and trial type

```
wcn_score.df <- wcn.data %>%
  dplyr::filter(TaskType == "immediate") %>%
  dplyr::group_by(LadlabID, Productivity, TrialType=as.factor(WithinOutsideIHC)) %>%
  dplyr::summarize(correct = sum(Accuracy, na.rm = TRUE), total=n())

wcn_model.df <- wcn_score.df %>%
  mutate(Productivity = factor(Productivity, levels = c(0,1)),
         TrialType = factor(WithinOutsideIHC, levels = c(0,1)))%>%
  mutate(IHC = as.integer(IHC),
         Highest_Contig_NN = as.integer(Highest_Contig_NN),
```

```
         LadlabID = factor(LadlabID))%>%
  mutate(IHC.c = as.vector(scale(IHC, center = TRUE, scale=TRUE)), #scale and center for model fit
         FHC.c = as.vector(scale(FHC, center = TRUE, scale=TRUE)),
         Highest_Contig_NN.c = as.vector(scale(Highest_Contig_NN, center = TRUE, scale=TRUE)))
```

How many trials do kids have beyond their IHC?

```
wcn.data %>%
  dplyr::filter(TaskType == "immediate") %>%
  dplyr::group_by(Productivity, WithinOutsideIHC)%>%
  dplyr::summarise(n = n())
```

```
## # A tibble: 4 x 3
## # Groups:   Productivity [?]
##   Productivity  WithinOutsideIHC     n
##   <fct>         <chr>            <int>
## 1 Nonproductive outside            295
## 2 Nonproductive within              49
## 3 Productive    outside            227
## 4 Productive    within             405
```

Highest contiguous NN

```
wcn.wide %<>%
  dplyr::mutate(TaskItem = as.numeric(as.character(TaskItem)))

unique.nn <- as.vector(unique(wcn.wide$LadlabID))
#get the task items from wcn
nextnums <- as.vector(unique(wcn.wide$TaskItem))

#this is a function that pulls out the largest number for which a participant had a correct consecutive
get_contiguous <- function(){
  contig <- data.frame()
  for (sub in unique.nn) {
    tmp <- wcn.wide %>%
      dplyr::select(LadlabID, Age, AgeGroup, TaskItem, immediate)%>%
      filter(LadlabID == sub,
             immediate == 0)%>%
      mutate(TaskItem = sort(TaskItem))
    if (length(tmp$LadlabID) == 0) {
      highest_contig = 86
      sub_contig <- data.frame(sub, highest_contig)
      contig <- bind_rows(contig, sub_contig)
    } else if (length(tmp$TaskItem) > 0 & min(tmp$TaskItem) == 23) {
      #if(sub %in% one.corr){
      highest_contig = 5
      sub_contig <- data.frame(sub, highest_contig)
      contig <- bind_rows(contig, sub_contig)
      # } else if(sub %in% five.corr | sub %in% zero.corr){
      #   highest_contig = 0
      #   sub_contig <- data.frame(sub, highest_contig)
      #   contig <- bind_rows(contig, sub_contig)
      # } else {
      #   highest_contig = 5
      #   sub_contig <- data.frame(sub, highest_contig)
```

```r
      #   contig <- bind_rows(contig, sub_contig)
      # }
    } else {
      min.nn <- min(tmp$TaskItem)
      prev_correct <- nextnums[nextnums < min.nn]
      highest_contig <- max(prev_correct)

      sub_contig <- data.frame(sub,
                               highest_contig)
      contig <- bind_rows(contig, sub_contig)
    }
  }
  return(contig)
}

highest_contiguous_nn <- get_contiguous()%>%
  dplyr::rename(LadlabID = sub)%>%
  distinct(LadlabID, highest_contig)%>%
  rename(Highest_Contig_NN = highest_contig)
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

```
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```

Code for checking highest contig NN

```
full.data %>%
  filter(LadlabID == "022316-AB") %>%
  filter(TaskType == "immediate"|TaskType == "practice") %>%
  select(LadlabID, TaskType, TaskItem, Accuracy)

# these two kids, for example, had the same contig highest NN but diff profile of responses
#040317-KK #7 correct out of 10
#022316-AB #9 correct out of 10


# wcn.data %<>%
#   dplyr::right_join(highest_contiguous_nn)
#
# wcn.data %>%
#   filter(LadlabID == "040317-KK")
```

See if highest contiguous next number underestimates kids' knowledge. Seems to correspond well with # correct data.

```
wcn.data %>%
  dplyr::right_join(highest_contiguous_nn) %>%
  dplyr::filter(TaskType == "immediate"|TaskType == "practice") %>% #added prac for 1&5
  dplyr::group_by(LadlabID, Highest_Contig_NN) %>%
  dplyr::summarize(n_corr = sum(Accuracy)) %>%
  dplyr::group_by(Highest_Contig_NN, n_corr, na.rm=TRUE) %>%
  dplyr::summarize(n_participants = n_distinct(LadlabID)) %>%
  tidyr::spread(n_corr, n_participants) %>%
  kable()
```

```
## Joining, by = "LadlabID"
```

Highest_Contig_NN	na.rm	1	2	3	4	5	6	7	8	9	10	
5	TRUE	1	13	11	7	NA	2	2	NA	NA	NA	NA
23	TRUE	NA	2	2	5	7	4	14	7	1	NA	1
29	TRUE	NA	NA	NA	NA	NA	NA	1	3	NA	NA	NA
37	TRUE	NA	NA	NA	NA	NA	1	1	3	2	NA	NA
40	TRUE	NA	NA	NA	NA	NA	NA	1	3	4	NA	NA
62	TRUE	NA	NA	NA	NA	NA	NA	NA	NA	2	NA	NA
70	TRUE	NA	NA	NA	NA	NA	NA	NA	NA	1	NA	NA
86	TRUE	NA	NA	NA	NA	NA	NA	NA	NA	NA	21	NA

```
# 2 kids had NA as n_corr
wcn.data %>%
  dplyr::right_join(highest_contiguous_nn) %>%
  dplyr::filter(TaskType == "immediate"|TaskType == "practice") %>% #added prac for 1&5
  dplyr::group_by(LadlabID, Highest_Contig_NN) %>%
  dplyr::summarize(n_corr = sum(Accuracy)) %>%
  dplyr::group_by(Highest_Contig_NN, n_corr) %>%
  filter(is.na(n_corr))
```

```
## Joining, by = "LadlabID"
```

```
## # A tibble: 1 x 3
## # Groups:   Highest_Contig_NN, n_corr [1]
##   LadlabID  Highest_Contig_NN n_corr
##   <chr>                 <dbl>  <int>
## 1 040317-SL                23     NA
```

```
# 022516-ML
# 040317-SL

# this kid (ML) got 0 for all test and NA for 1 and 5. Not one of the two kids under zero.corr because

# commentout
# full.data %>%
#   filter(LadlabID == "022516-ML") %>%
#   filter(TaskType == "immediate"|TaskType == "practice") %>%
#   select(LadlabID, TaskType, TaskItem, Accuracy)

# this kid (SL) has one NA value but otherwise look fine
# now added na.rm=TRUE for sum(accuracy)

# commentout
# full.data %>%
#   filter(LadlabID == "040317-SL") %>%
#   filter(TaskType == "immediate"|TaskType == "practice") %>%
#   select(LadlabID, TaskType, TaskItem, Accuracy)

# overview of highest contiguous coding and by-trial performance
z <- wcn.data %>%
  dplyr::right_join(highest_contiguous_nn) %>%
  filter(TaskType == "immediate") %>%
  select(LadlabID, Highest_Contig_NN, TaskItem, Accuracy) %>%
  spread(TaskItem, Accuracy)
```

```
## Joining, by = "LadlabID"
```

Median highest contiguous next number by productivity

```
full.data %>%
  dplyr::right_join(highest_contiguous_nn) %>%
  dplyr::distinct(LadlabID, Highest_Contig_NN, Productivity) %>%
  dplyr::group_by(Productivity) %>%
  dplyr::summarise(median_NN = median(Highest_Contig_NN),
                   mean_NN = mean(Highest_Contig_NN)) %>%
  kable()
```

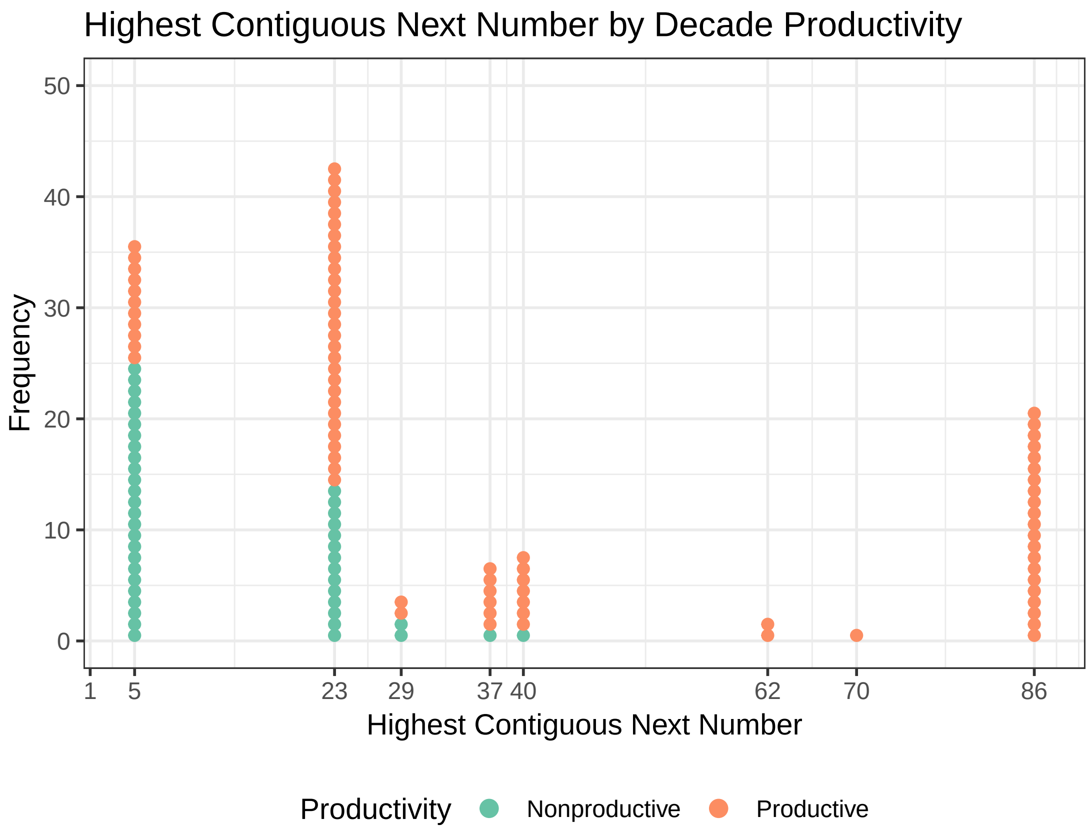
```
## Joining, by = "LadlabID"
```

Productivity	median_NN	mean_NN
Nonproductive	5	13.53488
Productive	23	41.54430

Plotting freq of highest contiguous as a function of productivity

```
full.data %>%
  dplyr::right_join(highest_contiguous_nn) %>%
  dplyr::distinct(LadlabID, Highest_Contig_NN, Productivity) %>%
  ggplot(aes(x=Highest_Contig_NN, color=Productivity)) +
  geom_dotplot(aes(fill = Productivity),
               binwidth=1, stackgroups=TRUE, binpositions="all",method="dotdensity") +
  scale_color_brewer(palette="Set2") +
  scale_fill_brewer(palette="Set2") +
  coord_fixed(ratio=1) +
  scale_y_continuous(breaks=seq(0,50,10), lim=c(0,50)) +
  #scale_x_continuous(breaks=seq(0,100,by=10)) +
  scale_x_continuous(breaks = c(0, 1, 5, 23, 29, 37, 40, 62, 70, 86),
                     labels=c("0", "1", "5", "23", "29", "37", "40", "62", "70", "86")) +
  labs(title="Highest Contiguous Next Number by Decade Productivity",
       x="Highest Contiguous Next Number",
       y="Frequency") +
  theme_bw() +
  theme(legend.position="bottom")
```

```
## Joining, by = "LadlabID"
```

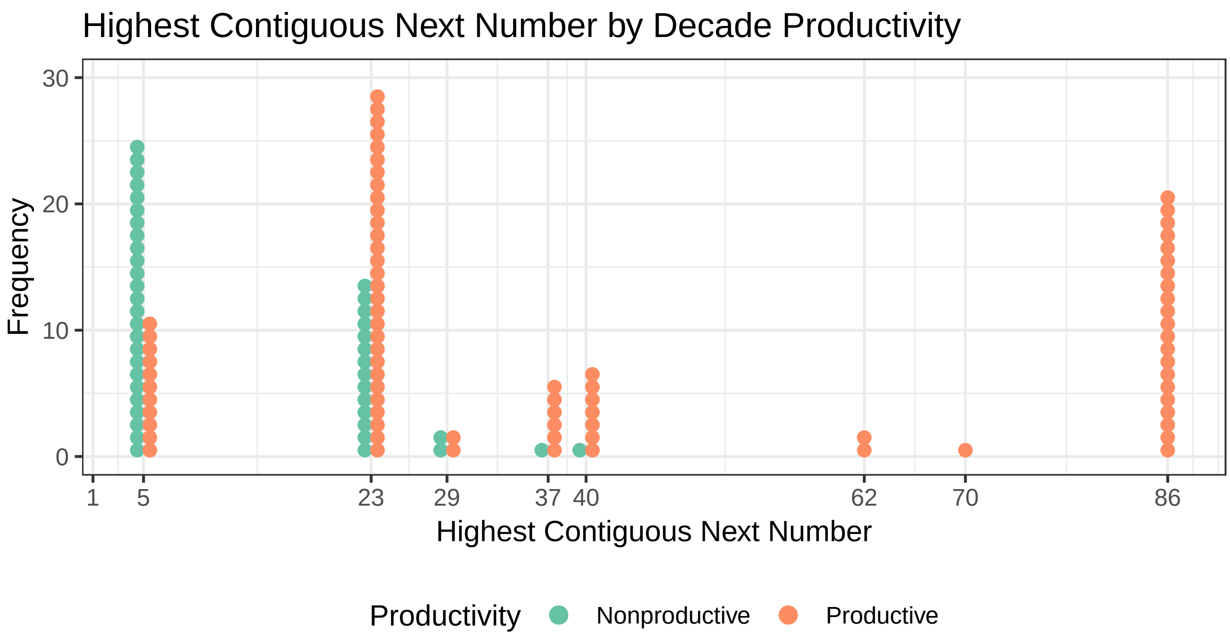



```r
ggsave('graphs/highestcontig-by-prod.png')
```

```
## Saving 6.5 x 4.5 in image
```

```r
# side by side
full.data %>%
  dplyr::right_join(highest_contiguous_nn) %>%
  dplyr::distinct(LadlabID, Highest_Contig_NN, Productivity) %>%
  ggplot(aes(x=Highest_Contig_NN, color=Productivity)) +
  geom_dotplot(aes(fill = Productivity),
               binwidth=1, stackdir="up", position=position_dodge(width=2), stackgroups=FALSE, binposit
  scale_color_brewer(palette="Set2") +
  scale_fill_brewer(palette="Set2") +
  coord_fixed(ratio=1) +
  scale_y_continuous(breaks=seq(0,30,10), lim=c(0,30)) +
  #scale_x_continuous(breaks=seq(0,100,by=10)) +
  scale_x_continuous(breaks = c(0, 1, 5, 23, 29, 37, 40, 62, 70, 86),
                     labels=c("0", "1", "5", "23", "29", "37", "40", "62", "70", "86")) +
  labs(title="Highest Contiguous Next Number by Decade Productivity",
       x="Highest Contiguous Next Number",
       y="Frequency") +
  theme_bw() +
  theme(legend.position="bottom")
```

```
## Joining, by = "LadlabID"
```




```r
ggsave('graphs/highestcontig-by-prod-2.png')
```

```
## Saving 6.5 x 4.5 in image
```

Correlations

```r
corrdf <- full.data %>%
  dplyr::right_join(highest_contiguous_nn) %>%
  dplyr::distinct(LadlabID, Highest_Contig_NN, Age, IHC, FHC) %>%
  dplyr::select(-LadlabID)
```

```
## Joining, by = "LadlabID"
```

```r
rcorr(as.matrix(corrdf), type = "pearson")
```

```
##                    Age  IHC  FHC Highest_Contig_NN
## Age               1.00 0.51 0.61              0.37
## IHC               0.51 1.00 0.72              0.75
## FHC               0.61 0.72 1.00              0.54
## Highest_Contig_NN 0.37 0.75 0.54              1.00
##
## n= 122
##
##
## P
##                   Age IHC FHC Highest_Contig_NN
## Age                   0   0   0
## IHC                0      0   0
## FHC                0   0      0
## Highest_Contig_NN  0   0   0
```

Infinity Descriptives

Number of kids in each infinity category

```r
full.data %>%
  dplyr::distinct(LadlabID, Category)%>%
  dplyr::group_by(Category)%>%
  dplyr::summarise(n = n())
```

```
## # A tibble: 4 x 2
##   Category              n
##   <fct>             <int>
## 1 A Non-knower         62
## 2 B Endless-only       14
## 3 C Successor-only     27
## 4 D Full-knower        19
```

```r
full.data %>%
  dplyr::distinct(LadlabID, SuccessorKnower, Productivity)%>%
  dplyr::group_by(SuccessorKnower, Productivity)%>%
  dplyr::summarise(n = n()) %>%
  spread(Productivity, n)
```

```
## # A tibble: 2 x 3
## # Groups:   SuccessorKnower [2]
##   SuccessorKnower Nonproductive Productive
##             <int>         <int>      <int>
## 1               0            31         39
## 2               1            12         40
```

```r
full.data %>%
  dplyr::distinct(LadlabID, EndlessKnower, Productivity)%>%
  dplyr::group_by(EndlessKnower, Productivity)%>%
  dplyr::summarise(n = n()) %>%
  spread(Productivity, n)
```

```
## # A tibble: 2 x 3
## # Groups:   EndlessKnower [2]
##   EndlessKnower Nonproductive Productive
##           <int>         <int>      <int>
## 1             0            39         49
## 2             1             4         30
```

Average age of kids for Endless and Successor Knowers

```r
full.data %>%
  dplyr::distinct(LadlabID, SuccessorKnower, Age)%>%
  dplyr::group_by(SuccessorKnower)%>%
  dplyr::summarise(meanAge = mean(Age),
                   sdAge = sd(Age),
                   meanAgeMonths = mean(Age)*12,
                   sdAgeMonths = sd(Age)*12)
```

```
## # A tibble: 2 x 5
##   SuccessorKnower meanAge sdAge meanAgeMonths sdAgeMonths
##             <int>   <dbl> <dbl>         <dbl>       <dbl>
## 1               0    4.92 0.577          59.0        6.93
## 2               1    5.11 0.550          61.3        6.60
```

```r
full.data %>%
  dplyr::distinct(LadlabID, EndlessKnower, Age)%>%
```

```
  dplyr::group_by(EndlessKnower)%>%
  dplyr::summarise(meanAge = mean(Age),
                   sdAge = sd(Age),
                   meanAgeMonths = mean(Age)*12,
                   sdAgeMonths = sd(Age)*12)
```

```
## # A tibble: 2 x 5
##   EndlessKnower meanAge sdAge meanAgeMonths sdAgeMonths
##           <int>   <dbl> <dbl>         <dbl>       <dbl>
## 1             0    4.89 0.560          58.7        6.71
## 2             1    5.27 0.516          63.2        6.19
```

Infinity in relation to highest count

```
full.data %>%
  dplyr::distinct(LadlabID, EndlessKnower, IHC, FHC) %>%
  dplyr::group_by(EndlessKnower) %>%
  dplyr::summarize(mean_IHC = mean(IHC),
                   mean_FHC = mean(FHC))
```

```
## # A tibble: 2 x 3
##   EndlessKnower mean_IHC mean_FHC
##           <int>    <dbl>    <dbl>
## 1             0     42.8     63.8
## 2             1     70.1     91.8
```

```
full.data %>%
  dplyr::distinct(LadlabID, SuccessorKnower, IHC, FHC) %>%
  dplyr::group_by(SuccessorKnower) %>%
  dplyr::summarize(mean_IHC = mean(IHC),
                   mean_FHC = mean(FHC))
```

```
## # A tibble: 2 x 3
##   SuccessorKnower mean_IHC mean_FHC
##             <int>    <dbl>    <dbl>
## 1               0     47.4     67.0
## 2               1     54.5     77.7
```

Infinity in relation to WCN

```
full.data %>%
  dplyr::right_join(highest_contiguous_nn) %>%
  dplyr::distinct(LadlabID, EndlessKnower, Highest_Contig_NN) %>%
  dplyr::group_by(EndlessKnower) %>%
  dplyr::summarize(mean_contig_nn = mean(Highest_Contig_NN),
                   median_contig_nn = median(Highest_Contig_NN))
```

```
## Joining, by = "LadlabID"
```

```
## # A tibble: 2 x 3
##   EndlessKnower mean_contig_nn median_contig_nn
##           <int>          <dbl>            <dbl>
## 1             0           26.2               23
## 2             1           45.7               37
```

```
full.data %>%
  dplyr::right_join(highest_contiguous_nn) %>%
  dplyr::distinct(LadlabID, SuccessorKnower, Highest_Contig_NN) %>%
```

```r
  dplyr::group_by(SuccessorKnower) %>%
  dplyr::summarize(mean_contig_nn = mean(Highest_Contig_NN),
                   median_contig_nn = median(Highest_Contig_NN))
```

```
## Joining, by = "LadlabID"
```

```
## # A tibble: 2 x 3
##   SuccessorKnower mean_contig_nn median_contig_nn
##             <int>          <dbl>            <dbl>
## 1               0           27.4               23
## 2               1           37.4               23
```

Analyses

Counting, Productivity, and Infinity Battery

To identify whether there is connection between counting experience and Infinity Task performance, we will conduct three initial analyses, predicting Infinity Task performance from either (1) Initial Highest Count, (2) Productivity for Decade Rule (defined above), or (3) performance on the Next Number task.

glmer(inf.0/1 ~ (predictor) + age + (1|subject), family = binomial).

First, we need to make a model data frame that readily has all of this information

```r
#model base
model.df <- full.data %>%
  dplyr::select(LadlabID, Age, AgeGroup, Gender, Task, Response, SuccessorKnower, EndlessKnower,
                IHC, FHC, DCE, Productivity)
```

Highest Next Number - commented because we're using highest contiguous

```r
# lookup <- full.data %>%
#   filter(Task == "WCN",
#          Accuracy == 1)%>%
#   group_by(LadlabID)%>%
#   summarise(max = max(as.numeric(as.character(TaskItem))))
#
#
# #Add highest NN to model df
# add_highest_num <- function() {
#   tmp <- model.df
#   for (row in 1:nrow(tmp)) {
#     sub = as.character(tmp[row, "LadlabID"])
#     highest_num = subset(lookup, LadlabID == sub)$max
#     tmp[row, "Highest_NN"] = highest_num
#   }
#   return(tmp)
# }
#
# #run this function on model df
# model.df <- add_highest_num()
```

Add highest contiguous next number to model.df

```r
model.df <- right_join(model.df, highest_contiguous_nn, by = "LadlabID")

# hc.datawide <- right_join(hc.datawide, highest_contiguous_nn, by = "LadlabID")

# hc.datawide %>%
#   dplyr::select(LadlabID, Age, AgeGroup, productivity, max,
#     HCReceivedSupport, ihc, dce, sup.noerror) %>%
#   group_by(productivity) %>%
#   summarize(median = median(max, na.rm=TRUE),
#             count = n())
# #median is 86 for all groups
```

Get mean WCN for everyone Not using this anymore - RMS

```r
# lookup <- wcn.wide %>%
#   group_by(LadlabID)%>%
#   summarise(mean.NN = mean(immediate, na.rm = TRUE))
```

Successor models

```r
#each participant only needs one row here, because we only need to know whether they are a Successor Kn
distinct_model.df <- model.df %>%
  distinct(LadlabID, Age, AgeGroup, Gender, SuccessorKnower, EndlessKnower,
           IHC, Highest_Contig_NN, FHC, DCE, Productivity)%>%
  mutate(SuccessorKnower = factor(SuccessorKnower, levels = c(0,1)),
         EndlessKnower = factor(EndlessKnower, levels = c(0,1)))%>%
  mutate(IHC = as.integer(IHC),
         Highest_Contig_NN = as.integer(Highest_Contig_NN),
         LadlabID = factor(LadlabID))%>%
  mutate(IHC.c = as.vector(scale(IHC, center = TRUE, scale=TRUE)), #scale and center for model fit
         FHC.c = as.vector(scale(FHC, center = TRUE, scale=TRUE)),
         Highest_Contig_NN.c = as.vector(scale(Highest_Contig_NN, center = TRUE, scale=TRUE)))

# #add mean_nn to model df
# distinct_model.df <- right_join(distinct_model.df, lookup, by = "LadlabID")

###MODEL BUILDING AND COMPARISONS###
#base model for successor knower
base.successor <- glmer(SuccessorKnower ~ Age + (1|LadlabID), family = "binomial",
                        data = distinct_model.df)

##IHC model##
model.ihc.successor <- glmer(SuccessorKnower ~ IHC.c + Age + (1|LadlabID), family = "binomial",
                             data = distinct_model.df)

#compare
anova(base.successor, model.ihc.successor, test = 'LRT') #IHC not significant
```

```
## Data: distinct_model.df
## Models:
## base.successor: SuccessorKnower ~ Age + (1 | LadlabID)
## model.ihc.successor: SuccessorKnower ~ IHC.c + Age + (1 | LadlabID)
##                     Df    AIC    BIC  logLik deviance  Chisq Chi Df
```

```
## base.successor       3 169.08 177.49 -81.539   163.08
## model.ihc.successor  4 171.01 182.22 -81.503   163.01 0.0711      1
##                     Pr(>Chisq)
## base.successor
## model.ihc.successor     0.7897
```

```
##Highest NN Model##
model.nn.successor <- glmer(SuccessorKnower ~ Highest_Contig_NN.c + Age + (1|LadlabID), family = "binomi
                           data = distinct_model.df)

#compare
anova(base.successor, model.nn.successor, test = 'LRT')#highest contiguous NN not significant
```

```
## Data: distinct_model.df
## Models:
## base.successor: SuccessorKnower ~ Age + (1 | LadlabID)
## model.nn.successor: SuccessorKnower ~ Highest_Contig_NN.c + Age + (1 | LadlabID)
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## base.successor      3 169.08 177.49 -81.539   163.08
## model.nn.successor  4 169.22 180.44 -80.610   161.22 1.8565      1
##                    Pr(>Chisq)
## base.successor
## model.nn.successor      0.173
```

```
##Productivity model##
model.prod.successor <- glmer(SuccessorKnower ~ Productivity + Age + (1|LadlabID), family = "binomial",
                             data = distinct_model.df)

#convergence warnings, is this an issue?
with(model.prod.successor@optinfo$derivs,max(abs(solve(Hessian,gradient)))<2e-3)#true, so we're okay
```

```
## [1] TRUE
```

```
#compare
anova(base.successor, model.prod.successor, test = 'LRT')#Productivity trending
```

```
## Data: distinct_model.df
## Models:
## base.successor: SuccessorKnower ~ Age + (1 | LadlabID)
## model.prod.successor: SuccessorKnower ~ Productivity + Age + (1 | LadlabID)
##                      Df    AIC    BIC  logLik deviance  Chisq Chi Df
## base.successor        3 169.08 177.49 -81.539   163.08
## model.prod.successor  4 167.95 179.17 -79.977   159.95 3.1234      1
##                      Pr(>Chisq)
## base.successor
## model.prod.successor    0.07718 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Endless Models

```
base.endless <- glmer(EndlessKnower ~ Age + (1|LadlabID), family = "binomial",
                      data = distinct_model.df)

###IHC MODEL###
```

```r
model.ihc.endless <- glmer(EndlessKnower ~ IHC.c + Age + (1|LadlabID), family = "binomial",
                          data = distinct_model.df)
#compare
anova(base.endless, model.ihc.endless, test = 'LRT') #IHC significant
```

```
## Data: distinct_model.df
## Models:
## base.endless: EndlessKnower ~ Age + (1 | LadlabID)
## model.ihc.endless: EndlessKnower ~ IHC.c + Age + (1 | LadlabID)
##                   Df    AIC    BIC  logLik deviance  Chisq Chi Df
## base.endless       3 139.31 147.72 -66.654   133.31
## model.ihc.endless  4 133.62 144.84 -62.810   125.62 7.6888      1
##                   Pr(>Chisq)
## base.endless
## model.ihc.endless   0.005556 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
###HIGHEST CONTIG NN MODEL###
model.nn.endless <- glmer(EndlessKnower ~ Highest_Contig_NN.c + Age + (1|LadlabID), family = "binomial"
                          data = distinct_model.df)

anova(model.nn.endless, base.endless, test = 'LRT')#Highest contig NN significant
```

```
## Data: distinct_model.df
## Models:
## base.endless: EndlessKnower ~ Age + (1 | LadlabID)
## model.nn.endless: EndlessKnower ~ Highest_Contig_NN.c + Age + (1 | LadlabID)
##                  Df    AIC    BIC  logLik deviance  Chisq Chi Df
## base.endless      3 139.31 147.72 -66.654   133.31
## model.nn.endless  4 135.77 146.98 -63.884   127.77 5.5405      1
##                  Pr(>Chisq)
## base.endless
## model.nn.endless    0.01858 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
###PRODUCTIVITY MODEL###
model.prod.endless <- glmer(EndlessKnower ~ Productivity + Age + (1|LadlabID), family = "binomial",
                            data = distinct_model.df)

#compare
anova(model.prod.endless, base.endless, test = 'LRT')#Prod significant
```

```
## Data: distinct_model.df
## Models:
## base.endless: EndlessKnower ~ Age + (1 | LadlabID)
## model.prod.endless: EndlessKnower ~ Productivity + Age + (1 | LadlabID)
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## base.endless        3 139.31 147.72 -66.654   133.31
## model.prod.endless  4 136.19 147.40 -64.094   128.19 5.1201      1
##                    Pr(>Chisq)
## base.endless
## model.prod.endless    0.02365 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# #okay with about mean NN
# model2.endless <- glmer(EndlessKnower ~ mean.NN + Age + (1|LadlabID), family = "binomial",
#                         data = distinct_model.df)
# anova(model2.endless, base.endless, test = 'LRT')#mean NN significant
```

Endless: Large model comparison

Put all significant Endless predictors into large model, run model comparison

```
##BASE MODEL WITH IHC
large.endless.base <- glmer(EndlessKnower ~ IHC.c + Age + (1|LadlabID),
                            family = "binomial", data = distinct_model.df)
##add highest contig
large.endless.nn <- glmer(EndlessKnower ~ Highest_Contig_NN.c + IHC.c + Age + (1|LadlabID),
                          family = "binomial", data = distinct_model.df)
#compare
anova(large.endless.base, large.endless.nn, test = 'LRT')#Apparently ns?
```

```
## Data: distinct_model.df
## Models:
## large.endless.base: EndlessKnower ~ IHC.c + Age + (1 | LadlabID)
## large.endless.nn: EndlessKnower ~ Highest_Contig_NN.c + IHC.c + Age + (1 | LadlabID)
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## large.endless.base  4 133.62 144.84 -62.810   125.62
## large.endless.nn    5 135.28 149.30 -62.642   125.28 0.3349      1
##                    Pr(>Chisq)
## large.endless.base
## large.endless.nn       0.5628
```

```
##Productivity v. Highest contiguous##
large.endless.prod <- glmer(EndlessKnower ~ Productivity + IHC.c + (1|LadlabID),
                            family = "binomial", data = distinct_model.df)
#compare
anova(large.endless.base, large.endless.prod, test = 'LRT')##Productivity is significant
```

```
## Data: distinct_model.df
## Models:
## large.endless.base: EndlessKnower ~ IHC.c + Age + (1 | LadlabID)
## large.endless.prod: EndlessKnower ~ Productivity + IHC.c + (1 | LadlabID)
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## large.endless.base  4 133.62 144.84 -62.810   125.62
## large.endless.prod  4 133.61 144.83 -62.806   125.61 0.0064      0
##                    Pr(>Chisq)
## large.endless.base
## large.endless.prod  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##ALL THREE TOGETHER
large.endless.full <- glmer(EndlessKnower ~ Productivity + Highest_Contig_NN.c + IHC.c + (1|LadlabID),
                            family = "binomial", data = distinct_model.df)

#convergence warnings, is this an issue?
with(large.endless.full@optinfo$derivs,max(abs(solve(Hessian,gradient)))<2e-3)#True, we're okay
```

```
## [1] TRUE
```

```
#compare
anova(large.endless.prod, large.endless.full, test = 'LRT') #n.s.
```

```
## Data: distinct_model.df
## Models:
## large.endless.prod: EndlessKnower ~ Productivity + IHC.c + (1 | LadlabID)
## large.endless.full: EndlessKnower ~ Productivity + Highest_Contig_NN.c + IHC.c +
## large.endless.full:     (1 | LadlabID)
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## large.endless.prod  4 133.61 144.83 -62.806   125.61
## large.endless.full  5 135.40 149.42 -62.702   125.40 0.2092      1
##                    Pr(>Chisq)
## large.endless.prod
## large.endless.full     0.6474
```