

# **Exploration and Exploitation in Parkinson's Disease: Behavioral Analyses**

Björn Meder

Martha Sterf

Charley M. Wu

Matthias Guggenmos

2025-02-05

# Table of contents

<b>1 Abstract</b>	<b>5</b>
<b>2 Intro</b>	<b>6</b>
<b>3 Experiment</b>	<b>7</b>
3.1 Materials and procedure . . . . .	7
3.2 Sample . . . . .	8
3.3 Questionnaires . . . . .	8
3.3.1 Beck's depression inventory II (BDI-II) . . . . .	8
3.3.2 Mini-Mental State Examination (MMSE) . . . . .	9
3.3.3 Hoehn-Yahr scale for severity of Parkinson . . . . .	9
<b>4 Behavioral data</b>	<b>10</b>
4.1 Sample characteristics . . . . .	12
4.2 Performance: Rewards by round . . . . .	13
4.3 Performance: Rewards by group . . . . .	16
4.4 Performance: Learning curves . . . . .	17
4.4.1 Performance: Role of physiological and cognitive assessments (BDI, MMSE, Hoehn-Yahr) . . . . .	18
4.5 Exploration vs. exploitation choices . . . . .	20
4.6 Spatial trajectories . . . . .	23
4.6.1 Types of choices . . . . .	24
4.6.2 Distance as function of previous reward . . . . .	26
4.7 Bonus round judgments . . . . .	32
4.7.1 Prediction error . . . . .	33
4.7.2 Prediction error and confidence . . . . .	33
4.7.3 Analysis of selected tiles . . . . .	35
<b>5 Appendix</b>	<b>40</b>
5.1 Distribution of BDI, MMSE, and YH in each group . . . . .	40
5.2 Performance as function of BDI, MMSE, and Hoehn-Yahr . . . . .	43
5.2.1 Performance as function of depression score (BDI) . . . . .	43
5.2.2 Performance as function of Mini-Mental State Examination (MMSE) . . . . .	44
5.2.3 Performance as function of Hoehn-Yahr (Parkinson patients only) . . . . .	44

```

# Housekeeping: Load packages and helper functions
# Housekeeping
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(message = FALSE)
knitr::opts_chunk$set(warning = FALSE)
knitr::opts_chunk$set(fig.align='center')
knitr::opts_chunk$set(prefer_html = TRUE)

options(knitr.kable.NA = '')

packages <- c('gridExtra', 'BayesFactor', 'tidyverse', "RColorBrewer", "lme4", "sjPlot", "ls"
lapply(packages, require, character.only = TRUE)

set.seed(0815)

# file with various statistical functions, among other things it provides tests for Bayes Fac
source('statisticalTests.R')

# Wrapper for brm models such that it saves the full model the first time it is run, otherwise
run_model <- function(expr, modelName, path='brm', reuse = TRUE) {
  path <- paste0(path,'/', modelName, ".brm")
  if (reuse) {
    fit <- suppressWarnings(try(readRDS(path), silent = TRUE))
  }
  if (is(fit, "try-error")) {
    fit <- eval(expr)
    saveRDS(fit, file = path)
  }
  fit
}

# Setting some plotting params
w_box          <- 0.2      # width of boxplot, also used for jittering points and lines
line_jitter     <- w_box / 2
xAnnotate      <- -0.3

# jitter params
jit_height     <- 0.01
jit_width      <- 0.05
jit_alpha      <- 0.6

# colors for age groups

```

```
groupcolors <- c("#1b9e77", "#d95f02", "#7570b3")
```

# 1 Abstract

We investigated how patients with Parkinson’s disease (PD) balance the explore-exploit trade-off using a spatially correlated bandit task, where the spatial structure of rewards facilitated value generalization (i.e., nearby options yield similar rewards). Participants were tested either shortly after taking Levodopa (L-Dopa) or just before their next scheduled dose. Patients with polyneuropathy served as a control group, comparable in age, depressive symptoms, and basic cognitive functioning.

We conducted both behavioral and computational analyses, revealing distinct behavioral patterns and computational signatures. PD patients on L-Dopa effectively balanced exploration and exploitation to obtain rewards, though not as efficiently as polyneuropathy patients. In stark contrast, participants off L-Dopa rarely exploited known high-value options and primarily explored novel ones. This overreliance on exploration impaired their ability to navigate the explore-exploit trade-off and maximize rewards.

To better understand the mechanisms underlying these behavioral differences, we employed a computational approach using the Gaussian Process Upper Confidence Bound (GP-UCB) model. This model integrates similarity-based generalization with two distinct exploration mechanisms: directed exploration, which seeks to reduce uncertainty about rewards, and random exploration, which introduces stochastic variability in choice behavior. Analysis of the model parameters showed that behavioral differences between the on- and off-medication conditions were primarily driven by differences in uncertainty-directed exploration, while the level of random exploration remained unchanged. Both PD groups exhibited lower generalization than the control group, which may underlie their overall poorer performance.

In sum, PD patients off L-Dopa favored exploration over exploitation, leading to impaired reward maximization, while those on medication balanced the explore-exploit trade-off more successfully.

## 2 Intro

A central distinction between different forms of exploration behavior is that *directed exploration* reflects the drive for knowledge about novel options, whereas *undirected exploration* refers to random variability in the choice process (Giron et al., 2023; Meder et al., 2021; Sadeghiyeh et al., 2020; Schulz et al., 2019; Wu et al., 2018).

# 3 Experiment

We investigated how patients with Parkinson’s disease (PD) manage the explore-exploit trade-off using a spatially correlated multi-armed bandit task. Participants accumulated rewards by selecting tiles (options) with normally distributed rewards. The spatial correlation between rewards facilitated generalization, allowing participants to adapt to the structure of the environment and balance exploring new options versus exploiting known high-reward options.

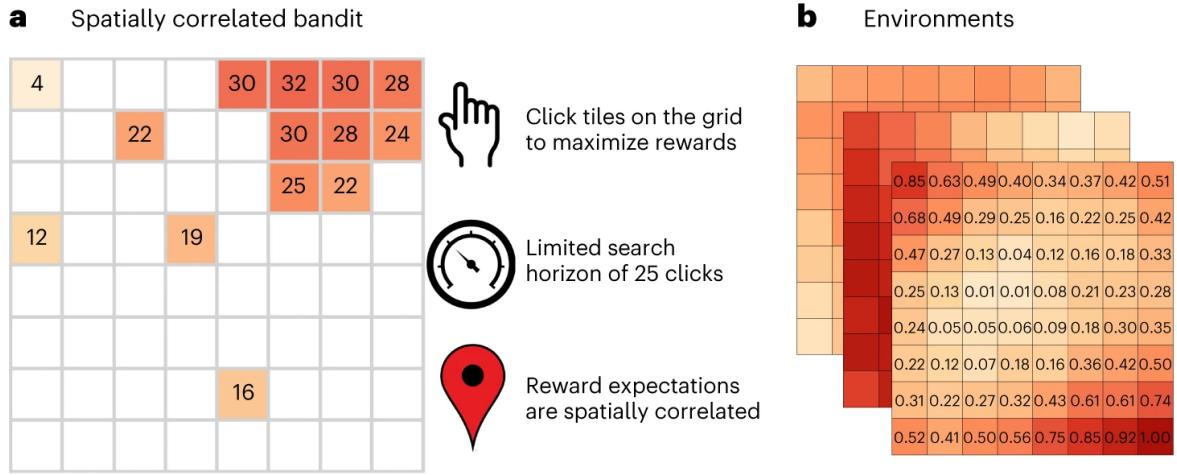


Figure 3.1: Screenshot from experiment and example environments (from Giron et al., 2023)

## 3.1 Materials and procedure

40 distinct environments were generated using a radial basis function kernel with  $\lambda = 4$ , creating a bivariate reward function on a grid that maps each tile location to a specific reward value. These smooth reward functions gradually varied across the grid.

Participants completed 10 rounds of the task, each featuring a new environment drawn without replacement from the set of 40 environments. In each round, participants had 25 choices to maximize rewards. The first round served as a tutorial to familiarize participants with the task and was excluded from the analyses. The final round (round 10) was a bonus round where, after 15 choices, participants were asked to predict rewards for five unrevealed options. Data from this round were also excluded from the main analysis and analyzed separately.

At the start of each round, one tile was randomly revealed, and participants sequentially sampled 25 tiles. On each trial, they could choose to either click a new tile or re-click a previously selected tile. Selections were made by selecting the tile on the computer screen, upon which they received a reward arbitrarily scaled to the range [0,50]. Re-clicked tiles showed small variations in reward due to normally distributed noise.

## 3.2 Sample

We collected data from adult participants with Parkinson's disease (PD) who receive Levodopa (L-Dopa) for treatment (Abbott, 2010; Tambasco et al., 2018). L-Dopa is a pharmacological agent used to treat the symptoms of Parkinson's disease, which is associated with a deficiency of dopamine. Since dopamine cannot pass the blood-brain barrier, L-Dopa is used as a precursor, which is absorbed through the gastrointestinal tract and subsequently metabolized to dopamine.

PD patients were randomly assigned to two conditions: on medication (PD+) and off medication (PD-). In the PD+ group ( $n=22$ ), L-Dopa was administered at least 30 minutes before the start of the experiment to allow the drug sufficient time to take effect during the task. In the PD- group ( $n=20$ ), the next scheduled dose for participants was timed such that they were in a low dopamine state during the experiment, offering a clear contrast to the PD+ group. Thus, we refer to the 'on medication' condition as the state after taking L-Dopa and the 'off medication' condition as the state before their next scheduled dose.

Patients with polyneuropathies (PNP) served as control group. The PNP group ( $n=20$ ) consisted of individuals with polyneuropathies, which is associated with physical symptoms similar to the motor impairments seen in Parkinson's disease. However, since polyneuropathy primarily affects peripheral nerves it is typically not associated with cognitive impairments, enabling a comparison in terms of physical symptomatology and the resulting burden of suffering.

Possible participants with Parkinson's disease were evaluated based on Hoehn-Yahr scores recorded in their patient files. The scale, which ranges from 1 to 5, assesses disease severity and motor impairments, with higher scores indicating greater severity (Goetz et al., 2004; Hoehn & Yahr, 1967). We limited recruitment to individuals with scores between 1 and 3, as scores of 4 and 5 reflect severe impairment.

## 3.3 Questionnaires

### 3.3.1 Beck's depression inventory II (BDI-II)

All participants answered the German version of the Beck Depression Inventory II, a self-report inventory consisting of 21 items measuring the severity of depression (Beck et al., 1996;

Hautzinger et al., 2006). Each item is scored from 0 to 3, with higher values representing more severe symptoms, such that the total score ranges between 0 and 63. A score of 0-11 is considered normal, 12-19 points indicate mild depressive symptoms, while values of 20 or higher suggest moderate to severe depression.

#### **i Note**

TO DO: check BDI scoring Wikipedia: Die Nationale Versorgungsleitlinie Unipolare Depression[3] listet im Anhang 1 S.217 (“Schwellenwerte bei psychometrischen Testverfahren”) folgende Grenzwerte für das BDI-II auf: <https://www.leitlinien.de/themen/depression> <https://doi.org/10.6101/AZQ%2F000496>

- 0–13: keine Depression bzw. klinisch unauffällig oder remittiert
- 14–19: leichtes depressives Syndrom
- 20–28: mittelgradiges depressives Syndrom
- 29: schweres depressives Syndrom

### **3.3.2 Mini-Mental State Examination (MMSE)**

The Mini-Mental State Examination (MMSE) is a clinical test used assess cognitive function and impairment, frequently used in patients with dementia (Folstein et al., 1975). The test comprises 30 questions pertaining to different domains, including as memory (e.g., recalling three objects), temporal and spatial orientation (e.g., date and location), arithmetic ability, and various others. Each correct answer receives one point, up to a maximum score of 30 points. In this study, an MMSE score between 27 and 30 points was defined as a criterion to establish a clear cutoff for excluding the presence of dementia or significant cognitive impairment, but none of the patients received such high scores.

### **3.3.3 Hoehn-Yahr scale for severity of Parkinson**

The Hoehn-Yahr scale assesses disease severity and motor impairments (Hoehn & Yahr, 1967). Participants can receive a score between one and five, with higher scores indicating more severe problems. We limited our recruitment to patients with scores ranging from 1 to 3.

## 4 Behavioral data

All behavioral data are stored in file *data\_gridsearch\_parkinson.csv*.

```
# read in data
dat      <- read_delim("data/data_gridsearch_parkinson.csv",
                       delim = ",",
                       col_types = cols(
                           #id = readr::col_factor(),
                           group = readr::col_factor(),
                           gender = readr::col_factor(),
                           z = col_double(),
                           zscaled = col_double(),
                           hoehn_yahr = col_double()
                       )) #>%>%
# arrange(as.numeric(as.character(id)))

# clean up
dat <- dat %>%
  select(-condition, -comments) %>%
  mutate(group = case_match(group,
                            "PPD-" ~ "PD-",
                            "PPD+" ~ "PD+",
                            "PNP" ~ "PNP",
                            .default = NA)) %>%
  mutate(group = factor(group, levels = c("PNP", "PD+", "PD-"))) %>%
  mutate(type_choice = factor(type_choice, levels = c("Repeat", "Near", "Far"))) %>%
  mutate(gender = recode(gender, "w" = "f")) %>%
  rename(MMSE = `mini_mental`) %>%
  mutate(last_ldopa = if_else(group != "PNP", as_hms(last_ldopa), as_hms(NA)),
         next_ldopa = if_else(group != "PNP", as_hms(next_ldopa), as_hms(NA)),
         time_exp = if_else(group != "PNP", as_hms(time_exp), as_hms(NA))) %>%
  mutate(time_since_ldopa = as.numeric(time_exp - last_ldopa, unit = "mins"))

# get subject information
df_sample <- dat %>%
  select(id, age, gender, group, BDI, MMSE, hoehn_yahr, last_ldopa, next_ldopa, time_exp, time_since_ldopa)
```

Table 4.1: Behavioral data.

id	session	x	y	chosen	z	zscaled	time	trial	round	distance	type_choice	previous_r
115	1	4	4	37	20	18	1.72e+17	0	1			
115	1	4	2	21	41	31	1.72e+17	1	1	2	Far	
115	1	3	2	20	45	34	1.72e+17	2	1	1	Near	
115	1	6	3	31	29	24	1.72e+17	3	1	4	Far	
115	1	3	1	12	38	29	1.72e+17	4	1	5	Far	
115	1	2	1	11	37	29	1.72e+17	5	1	1	Near	

```

group_by(id) %>%
slice_head(n = 1) %>%
arrange(group)

if (knitr:::is_html_output()) {
  head(dat) %>%
    kable("html", caption = "Behavioral data.") %>%
    kable_styling(bootstrap_options = c("striped", "hover", "condensed"), full_width = FALSE,
                  scroll_box(width = "100%", height = "300px"))
} else {
  head(dat) %>%
    kable("latex", caption = "Behavioral data.")
}

```

The data frame *dat* with the raw data contains the following variables:

- *id*: participant id
- *age* is participant age in years
- *gender*: (m)ale, (f)emale, (d)iverse
- *x* and *y* are the sampled coordinates on the grid
- *chosen*: are the *x* and *y* coordinates of the chosen tile
- *z* is the reward obtained from the chosen tile. Re-clicked tiles could show small variations in the observed color (i.e., underlying reward) due to normally distributed noise,  $\epsilon \sim N(0, 1)$ .
- *z\_scaled* is the observed outcome (reward), scaled in each round to a randomly drawn maximum value in the range of 70% to 90% of the darkest reward value
- *trial* is the trial number (0-25), with 0 corresponding to the initially revealed random tile.
- *round* is the round number (1 through 10), with 1=practice round (not analyzed) and 10=bonus round (analyzed only for bonus round judgments)
- *distance* is the Manhattan distance between consecutive clicks. *NA* for trial 0, i.e., initially revealed random tile.

- *type\_choice* categorizes consecutive clicks as “repeat” (clicking the same tile as in the previous round), “near” (clicking a directly neighboring tile, i.e. distance=1), and “far” (clicking a tile with distance > 1). *NA* for trial 0, i.e., the initially revealed random tile.
- *previous\_reward* is the reward *z* obtained on the previous step. *NA* for trial 0, i.e., the initially revealed random tile.
- *last\_ldopa*: time of the last L-Dopa dose (HH:MM)
- *next\_ldopa*: scheduled time of the next L-Dopa dose (HH:MM)
- *time\_exp*: time of the experiment (HH:MM)
- *time\_since\_ldopa*: time since last L-Dopa (in minutes)
- *wearing\_off\_min*:

## 4.1 Sample characteristics

We collected data from 62 adult participants with Parkinson disease, on and off L-Dopa medication, and Polyneuropathy. Let’s have a look at the sample:

```
df_sample %>%
  group_by(group) %>%
  summarise(n = n(),
            female = sum(gender == "f"),
            mean_age = mean(age),
            sd_age = sd(age),
            mean_BDI = mean(BDI, na.rm= T),
            mean_MMSE = mean(MMSE, na.rm= T),
            mean_HY = mean(hoehn_yahr, na.rm= T),
            mean_time_since_ldopa = mean(time_since_ldopa, na.rm= T)) %>%
{
  if (knitr:::is_html_output()) {
    kable(., format = "html", escape = FALSE, digits = 1) %>%
      kable_styling("striped", full_width = FALSE)
  } else {
    kable(., format = "latex", digits = 1)
  }
}
```

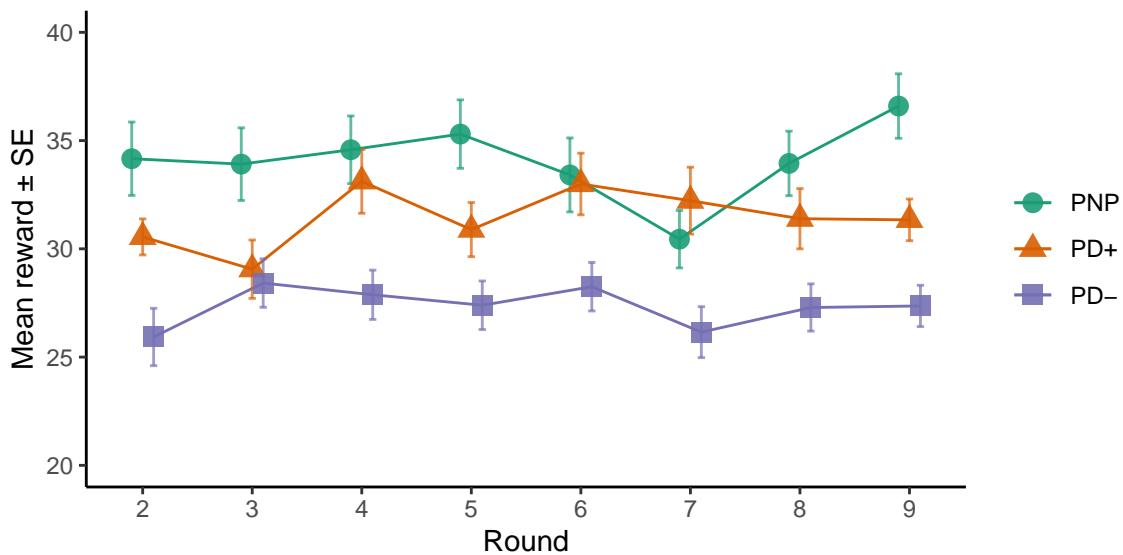
group	n	female	mean_age	sd_age	mean_BDI	mean_MMSE	mean_HY	mean_time_since_ld
PNP	20	13	63.9	8.5	8.1	29.1		
PD+	22	11	61.6	5.8	8.0	29.2	1.9	1
PD-	20	9	65.8	7.9	7.2	29.0	2.0	2

We analyzed the behavioral data in terms of performance and exploration behavior. These analyses exclude the tutorial and bonus rounds, leaving a total of 200 search decisions (8

rounds  $\times$  25 trials) for each participant. We then report the results of the bonus round, where we analyze participants' reward predictions and confidence judgments. We report both frequentist statistics and Bayes factors ( $BF$ ) to quantify the relative evidence of the data in favor of the alternative hypothesis ( $H_A$ ) over the null hypothesis ( $H_0$ ); see Appendix for details and references. Various helper functions are implemented in *statisticalTests.R*. Regression analyses were performed in a Bayesian framework with *Stan*, accessed via R-package *brms*, complemented by frequentist hierarchical regression analyses (via package *lmer*).

## 4.2 Performance: Rewards by round

First, let's plot the obtained rewards as function of group and round.



ANOVA results with *round* as within- and *group* as between-subjects factor show a difference between groups, with PNP patients achieving the greatest rewards, followed by PD+ and PD- patients, but no change across rounds (and no interaction). A Bayesian regression analysis yield the same results. Therefore, for the subsequent analyses we collapse across rounds.

```
aov_rounds <- aov_ez(
  id = "id",
  dv = "mean_reward",
  within = "round",
  between = "group",
  data = df_mean_reward_subject_by_round
)
```

Table 4.2: ANOVA results with round as within-subjects factor and group as between subjects factor, where rewards per round were first aggregated within subjects.

	num Df	den Df	MSE	F	ges	Pr(>F)
group	2.00	59.0	86.59	21.17	0.17	0.00
round	6.10	359.7	34.93	1.31	0.02	0.25
group:round	12.19	359.7	34.93	1.30	0.03	0.22

```
if (knitr::is_html_output()) {
  kable(as.data.frame(aov_rounds$anova_table),
    format = "html", escape = FALSE, digits = 2,
    caption = "ANOVA results with round as within-subjects factor and group as between subjects",
    kable_styling("striped", full_width = FALSE)
} else {
  kable(as.data.frame(aov_rounds$anova_table),
    format = "latex", digits = 2,
    caption = "ANOVA results with round as within-subjects factor and group as between subjects")
}
```

```
brm_rounds <- run_model(brm(
  mean_reward ~ round * group + (1 | id),    # Random intercept for subject
  data = df_mean_reward_subject_by_round,
  family = gaussian(),
  iter = 4000,
  warmup = 1000,
  chains = 4,
  cores = 4,
  seed = 0511),
  modelName = 'brm_reward_rounds')

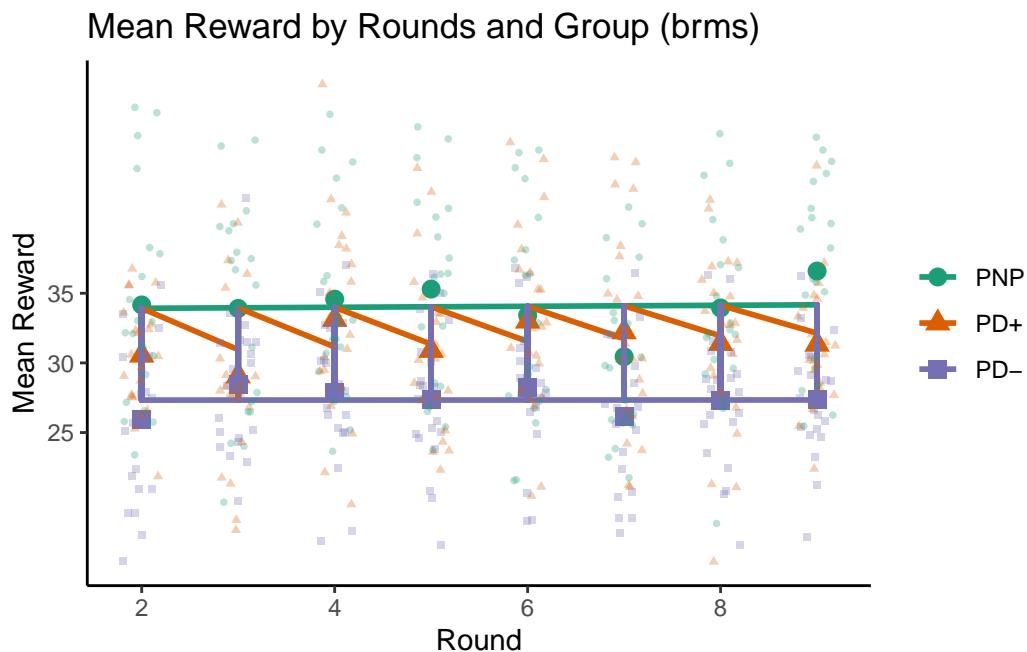
# Extract fitted values and add to data df
fitted_values <- fitted(brm_rounds, re_formula = NA)
df_mean_reward_subject_by_round$fitted_mean_reward <- fitted_values[, "Estimate"]

ggplot(df_mean_reward_subject_by_round, aes(x = round, y = mean_reward, group = group, shape = group),
  geom_point(data = df_summary_by_round, aes(x = round, y = mean_of_means, shape = group), size = 1) +
  geom_line(aes(y = fitted_mean_reward), size = 1) +
  geom_jitter(aes(x = round, y = mean_reward), size = 1, alpha = 0.3, width = 0.2) +
  scale_y_continuous("Mean Reward", breaks = c(25,30,35)) +
  xlab("Round") +
  scale_fill_manual(values = groupcolors) +
```

```

scale_color_manual(values = groupcolors) +
ggtitle("Mean Reward by Rounds and Group (brms)") +
theme_classic() +
theme(legend.title = element_blank())

```

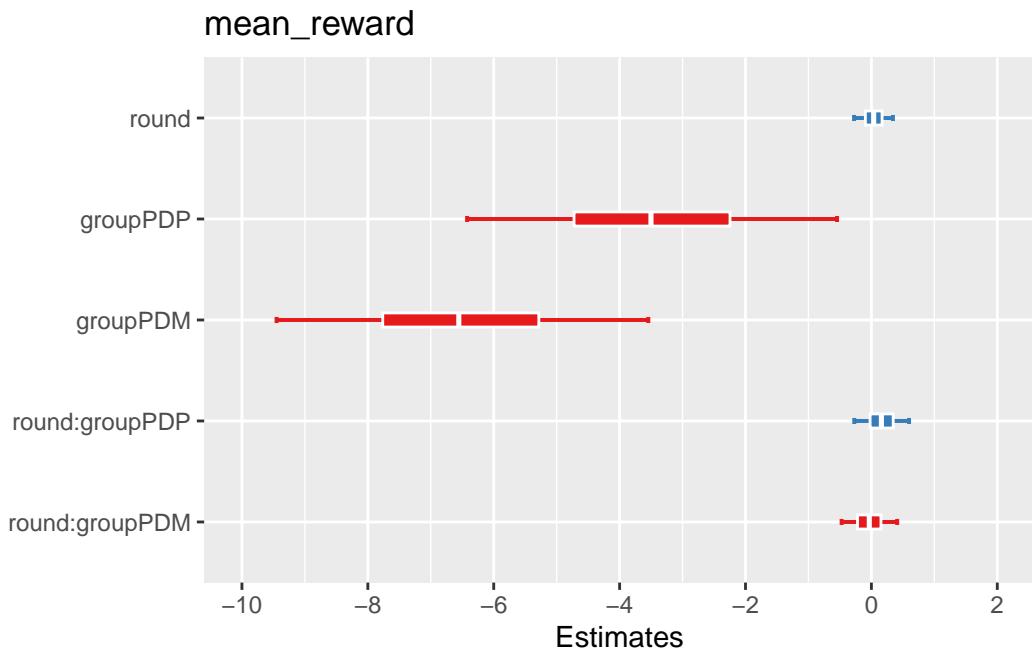


```

# tbl_regression(brm_rounds, exponentiate = F)
tab_model(brm_rounds)

```

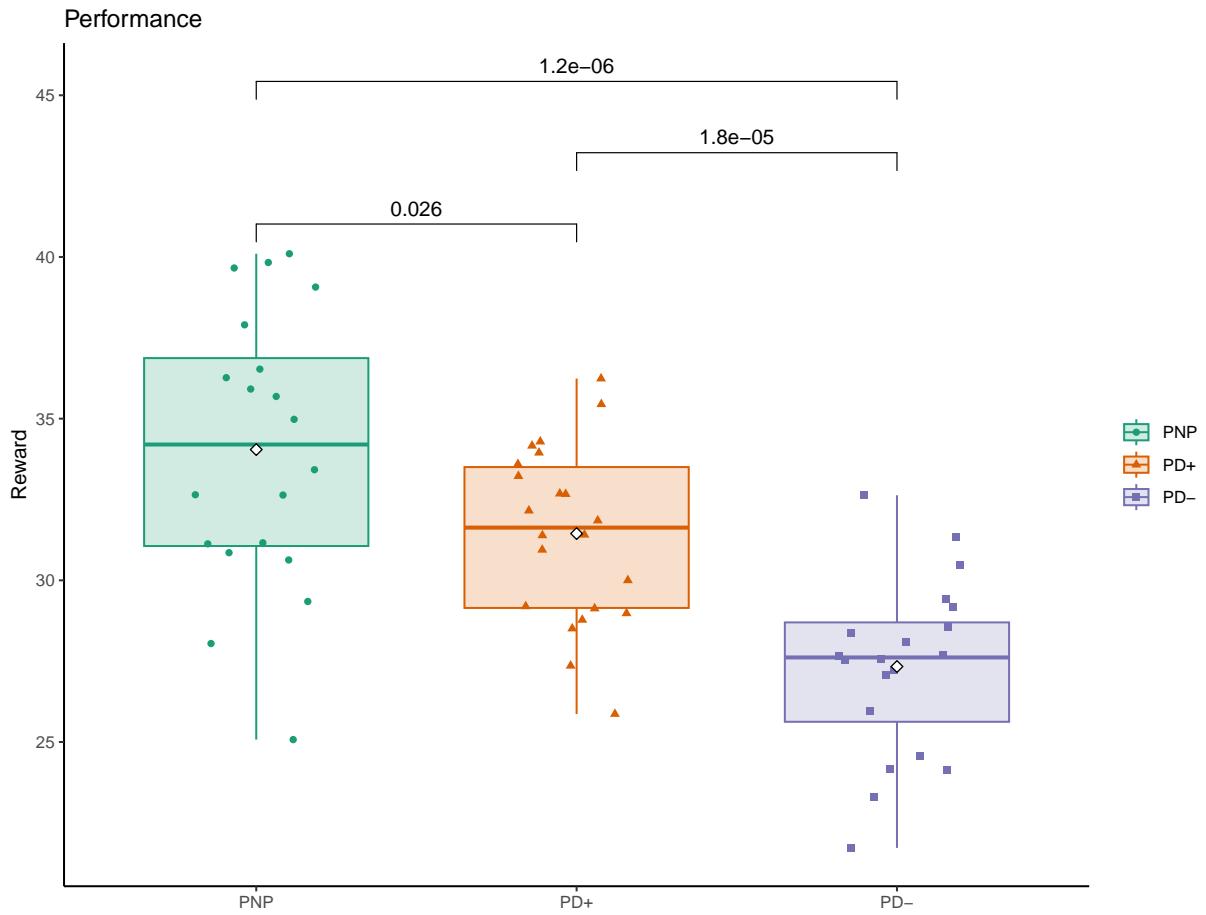
```
plot_model(brm_rounds, type = "est")
```



### 4.3 Performance: Rewards by group

The plot shows the overall performance of each group, where we compute for each subjects the mean reward across all trials. Each dot is one participants' mean reward across all rounds and trials.

group	n	m_reward	md_reward	var_reward	sd_reward	se_reward	lower_ci_reward	upper_ci_reward
PNP	20	34.04	34.20	17.97	4.24	0.95	32.06	
PD+	22	31.45	31.63	7.36	2.71	0.58	30.24	
PD-	20	27.33	27.61	7.50	2.74	0.61	26.05	



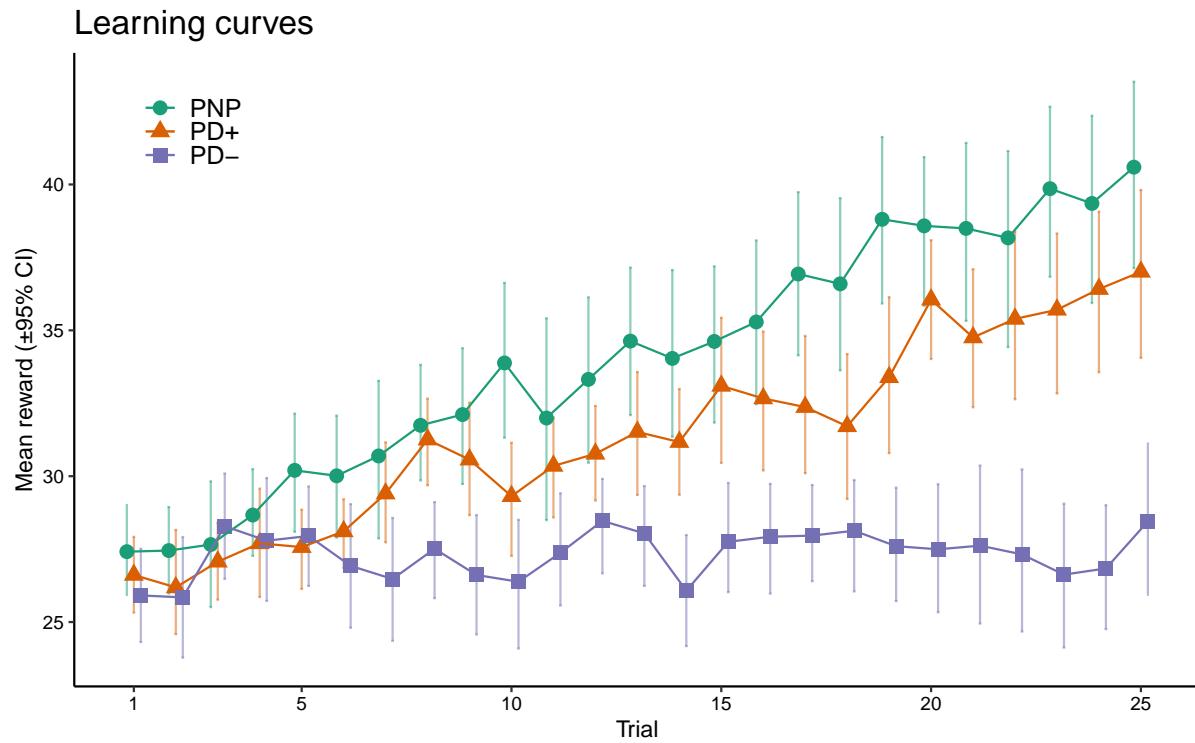
All groups differed, with PNP achieving the greatest amount of reward, followed by PD+ and PD-.

- PNP vs. PD+:  $t(40) = 2.4, p = .022, d = 0.7, BF = 2.7$
- PNP vs. PD-:  $t(38) = 5.9, p < .001, d = 1.9, BF > 100$
- PD+ vs. PD-:  $t(40) = 2.4, p = .022, d = 0.7, BF = 2.7$

#### 4.4 Performance: Learning curves

Participants' learning curves illustrate the average reward obtained in each trial across rounds. For both polyneuropathy patients (PNP) and Parkinson's patients on medication (PD+), the mean rewards increased as the round progresses, suggesting they effectively balanced exploration and exploitation to maximize rewards. In stark contrast, Parkinson's patients off medication (PD-) showed no improvement across trials.

TO DO: Add random reward as baseline



#### 4.4.1 Performance: Role of physiological and cognitive assessments (BDI, MMSE, Hoehn-Yahr)

We also assessed patients in terms of their depressive symptoms (via BDI-II), cognitive functioning (via Mini-Mental-Status Examination, MMSE), and severity of motor symptoms (via Hoehn-Yahr scale, Parkinson's disease patients only).

We ran a hierarchical regression with reward as dependent variable and group, BDI score, and MMSE score; with random intercepts for participants to account for individual differences. This analysis yielded only an effect of group, suggesting that BDI and MMSE score were not related to performance.

```
# Hierarchical frequentist regression with random intercept: Reward as function of BDI and MMSE
lmer_performance_BDI_MMSE <- lmer(z ~ group + BDI + MMSE + (1 | id),
                                     data = subset(dat, trial > 0 & round %in% 2:9))

#summary(lmer_reward_BDI_MMSE)

tab_model(lmer_performance_BDI_MMSE, title = "Hierarchical regression results: Performance as a function of BDI and MMSE")
```

```

# Hierarchical Bayesian regression with random intercept: Reward as function of BDI and MMSE
brm_performance_BDI_MMSE <- run_model(brm(z ~ group + BDI + MMSE + (1|id),
                                            data=subset(dat, trial > 0 & round %in% 2:9 ),
                                            cores=4,
                                            seed = 0815,
                                            iter = 5000,
                                            warmup=1000,
                                            control = list(adapt_delta = 0.99, max_treedepth = 15),
#prior = prior(normal(0,10), class = "b")),
                                            modelName = 'brm_performance_BDI_MMSE')
#tab_model(brm_performance_assessment, bpe="mean", title = "Hierarchical Bayesian regression")
#bayes_R2(brm_performance_assessment)
#tab_model(lmer_performance_BDI_MMSE, brm_performance_BDI_MMSE, title = "Hierarchical regressi

```

Next, we ran a hierarchical regression for Parkinson's patients only, with reward as dependent variable and group, BDI, MMSE, and Hoehn-Yahr score as predictors; with random intercepts for participants to account for individual differences. This analysis only yielded an influence of group, i.e. being on or off L-Dopa.

```

# Hierarchical frequentist regression with random intercept: Reward as function of BDI, MMSE
lmer_reward_PD_only_BDI_MMSE_HY <- lmer(z ~ group + BDI + MMSE + hoehn_yahr + (1 | id),
                                             data = subset(dat, trial > 0 & round %in% 2:9 & group ==
#summary(lmer_reward_PD_only_BDI_MMSE_HY)

tab_model(lmer_reward_PD_only_BDI_MMSE_HY, title = "Hierarchical regression results: Performance

# Hierarchical Bayesian regression with random intercept: Reward as function of BDI, MMSE, and HY
brm_performance_PD_only_BDI_MMSE_HY <- run_model(brm(z ~ group + BDI + MMSE + (1|id),
                                                       data=subset(dat, trial > 0 & round %in% 2:9 & group == 1),
                                                       cores=4,
                                                       seed = 0815,
                                                       iter = 5000,
                                                       warmup=1000,
                                                       control = list(adapt_delta = 0.99)),
                                                       #prior = prior(normal(0,10), class = "b")),
                                                       modelName = 'brm_performance_PD_only_BDI_MMSE_HY')

# tab_model(lmer_reward_PD_only_BDI_MMSE_HY, brm_performance_PD_only_BDI_MMSE_HYtitle = "Hier

```

## 4.5 Exploration vs. exploitation choices

As a global measure of exploration behavior, we calculated the proportion of unique tiles (out of 25) selected during each round. Higher proportions indicate a greater tendency to explore novel options, while lower proportions suggest a preference for exploiting known options. There were substantial differences between groups, with PD+ patients showing higher levels of exploratory behavior and, conversely, a reduced tendency to exploit known options. This effect was particularly pronounced for PD- patients, who almost exclusively selected novel options during the task.

```
# proportion of unique choices per round per subject
df_unique_choices_round <-
  dat %>%
  filter(round %in% 2:9 & trial > 0) %>%
  group_by(id, group, round) %>%
  summarize(
    total = n(), # number of trials
    unique_tiles = n_distinct(x, y), # unique (x, y) combinations (i.e., tiles)
    repeat_tiles = total - unique_tiles
  ) %>%
  mutate(prop_unique = unique_tiles/total,
        prop_repeat = repeat_tiles/total)

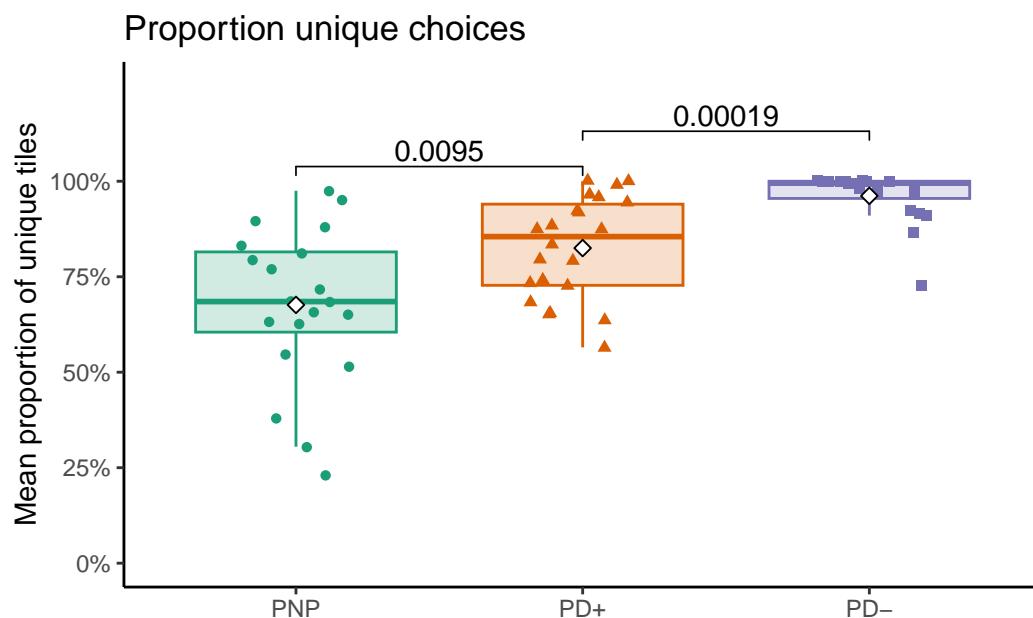
# proportion of unique choices across 8 rounds per subject
df_unique_choices_subject <- df_unique_choices_round %>%
  group_by(id, group) %>%
  summarize(m_prop_unique = mean(prop_unique),
            m_prop_repeat = mean(prop_repeat))

ggbboxplot(df_unique_choices_subject,
           x = "group",
           y = "m_prop_unique",
           color = "group", palette = groupcolors, fill = "group", alpha = 0.2,
           add = "jitter", jitter.size = 0.5, shape = "group", title = "Proportion unique choices per subject",
           scale_y_continuous("Mean proportion of unique tiles",
                             breaks = c(0, 0.25, 0.5, 0.75, 1),
                             labels = percent_format(accuracy = 1)) +
  coord_cartesian(ylim=c(0,1.25)) +
  xlab("") +
  stat_compare_means(comparisons = list( c("PNP", "PD+"), c("PD+", "PD-") ) ,
                     paired = F,
```

```

        method = "t.test",
        # label = "p.format",
        aes(label = paste0("p = ", after_stat(p.format)))
    ) +
stat_summary(fun = mean, geom="point", shape = 23, fill = "white", size=2) +
theme_classic() +
theme(strip.background = element_blank(),
strip.text = element_text(color = "black", size=12),
legend.position = "none"
)

```



```
ggsave("plots/unique_choices.png", dpi=300, width =5, height=4.5)
```

To investigate the temporal dynamics of exploration and exploitation, we determined for each trial whether the chosen tile was novel (an exploration decision) or had already been selected previously (an exploitation decision). The data reveal that the likelihood of exploiting a previously chosen option increased significantly over time for both PNP and PD+ patients, indicating a gradual shift in behavior from exploration to exploitation as the round progressed. In contrast, PD- patients predominantly engaged in exploration throughout the round and showed only a weak tendency towards increased exploitation as the search horizon approached its end.

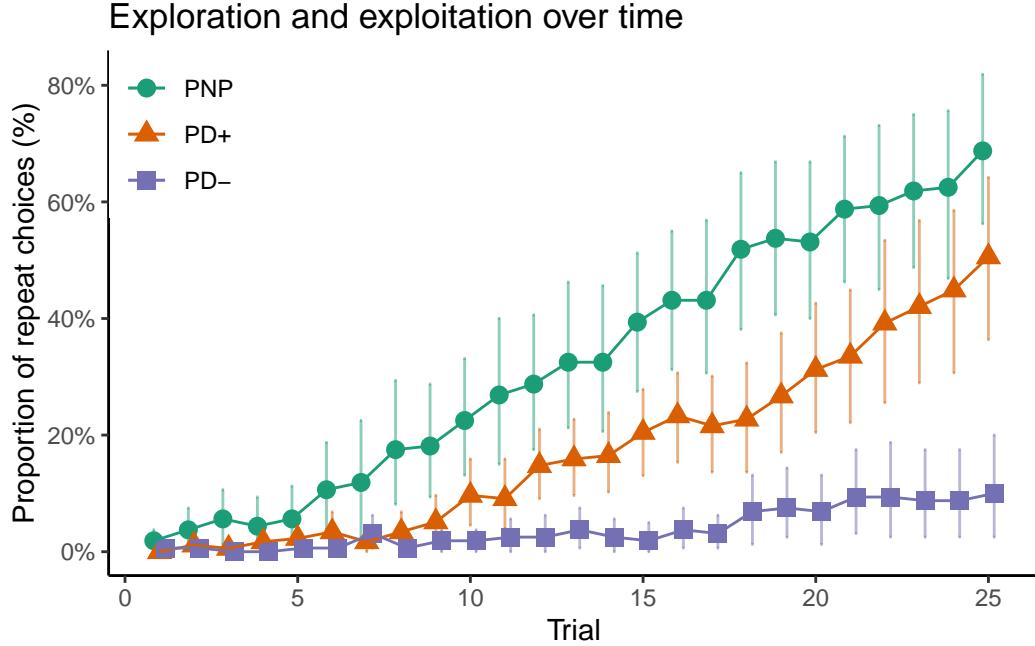
```

dat <- dat %>%
  group_by(id, round) %>%
  arrange(trial, .by_group = TRUE) %>% # Ensure data is sorted by trial
  mutate(
    is_new = factor(if_else(!duplicated(chosen), "new", "repeat")) # Check uniqueness based
  ) %>%
  ungroup()

dat_repeat_prop <- dat %>%
  filter(trial > 0 & round %in% 2:9) %>%
  group_by(id, group, trial) %>%
  summarize(
    prop_repeat = mean(is_new == "repeat", na.rm = TRUE) # Calculate proportion of "repeat"
  )

# plot proportion of repeat choices across trials
ggplot(dat_repeat_prop, aes(x = trial, y = prop_repeat, fill = group, shape = group, color =
  stat_summary(fun.data = mean_cl_boot, geom = "errorbar", width = 0.2, alpha = 0.5, , position
  stat_summary(fun = mean, geom = "point", size = 3, position=position_dodge(width=0.5)) +
  stat_summary(fun = mean, geom = "line", position=position_dodge(width=0.5)) +
  scale_fill_manual(values = groupcolors) +
  scale_color_manual(values = groupcolors) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(
    x = "Trial",
    y = "Proportion of repeat choices (%)",
    title = "Exploration and exploitation over time"
  ) +
  theme_classic() +
  theme(strip.background = element_blank(),
        strip.text = element_text(color = "black", size=12),
        legend.position = "inside",
        legend.position.inside = c(0.15, 1), # Use legend.position.inside
        legend.justification = c(1, 1),
        legend.title = element_blank()
  )
)

```

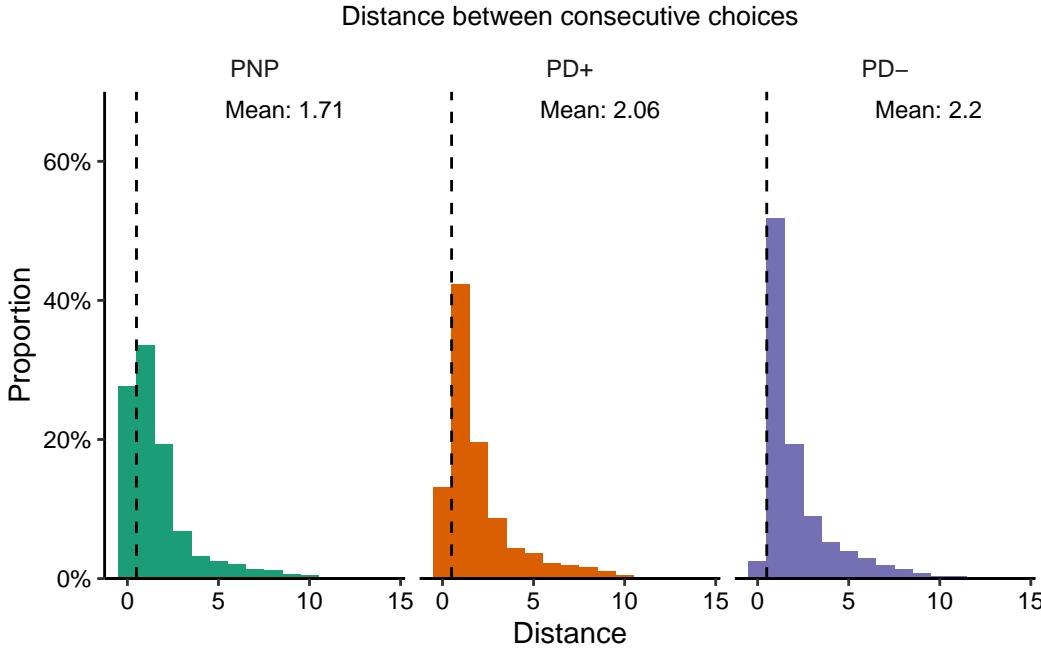


```
ggsave("plots/explore-exploit_time.png", dpi=300, width = 6, height =4)
```

## 4.6 Spatial trajectories

We next consider participant's spatial search trajectories (distance among consecutive clicks). Distance is measured as Manhattan distance between consecutive clicks, such that repeat clicks have distance 0, clicking directly neighbouring tiles has distance 1, and clicks further away have distances  $>1$ .

The most frequent choice was to select a neighboring tile (distance = 1), reflecting a local search approach (Wu et al., 2018). On average, PNP patients had the shortest distances, indicating more local searches and repeated clicks. PD+ patients had greater distances than PNP but shorter than PD- patients, who showed the highest distances. The distribution of distances shows that this is primarily due to the few repeat choices (distance = 0) they made, i.e. very limited exploitation behavior.

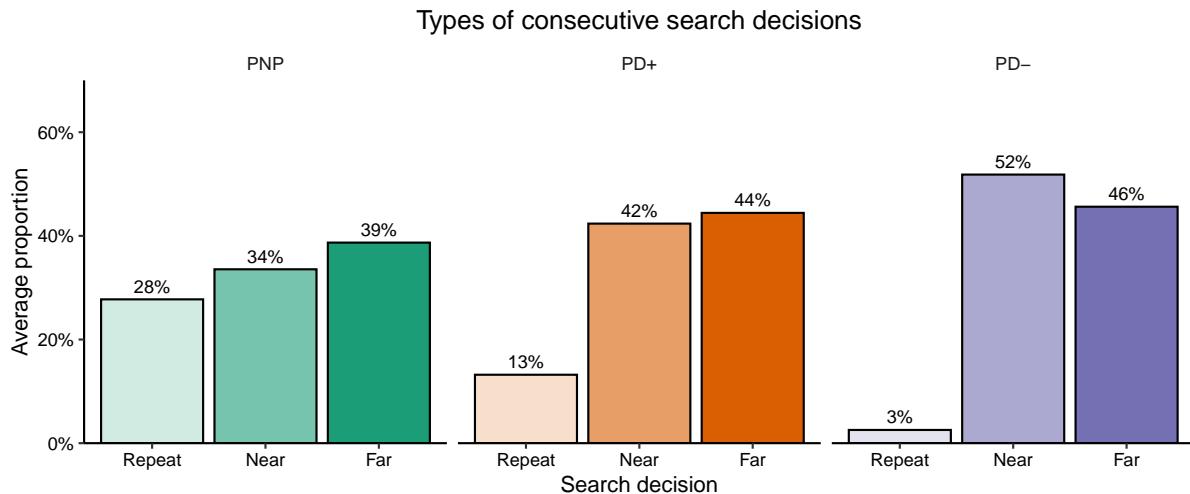


PNP patients had lower search distances than the PD+ group:  $t(40) = -2.2, p = .035, d = 0.7, BF = 1.9$  and lower distances than the PD- group,  $t(40) = -2.2, p = .035, d = 0.7, BF = 1.9$ . There was no difference between Parkinson patient with (PD+) and without (PD-) medication,  $t(40) = -0.8, p = .454, d = 0.2, BF = .38$

#### 4.6.1 Types of choices

We can also categorize each consecutive click as “repeat” (clicking the same tile as in the previous round), “near” (clicking a directly neighboring tile, i.e. distance=1), or “far” (clicking a tile with distance  $> 1$ ). We first computed for each participant the proportion of type of choices across all 8 rounds x 25 clicks = 200 search decisions and then plot the mean proportion for each group.

The analyses reveal distinct search patterns across patient groups. PNP participants had the highest proportion of repeat (exploit) decisions, followed by the PD+ group. The proportion of repeat decisions in the PD- group was minimal. These behaviors help explain the differences in learning curves, where PNP patients showed the most significant improvement, followed by PD+ patients. In contrast, PD- patients exhibited no improvement across trials, due to their lower tendency to exploit high-reward options.



An analysis of consecutive choice types over time reveals clear differences in search behavior between the groups. Both PNP and PD+ patients adapt their strategies as the round progresses by decreasing the number of local (distance = 1) and far (distance > 1) choices, while increasing the number of exploit decisions, indicating a shift from exploration to exploitation. Notably, the data indicate a faster shift to exploitation for PNP patients compared to PD+ patients, with an earlier and stronger preference for re-selecting known high-reward options. In contrast, PD- patients show limited adaptation, with the proportions of each decision type remaining relatively stable throughout the round, aside from a slight increase in exploit decisions.

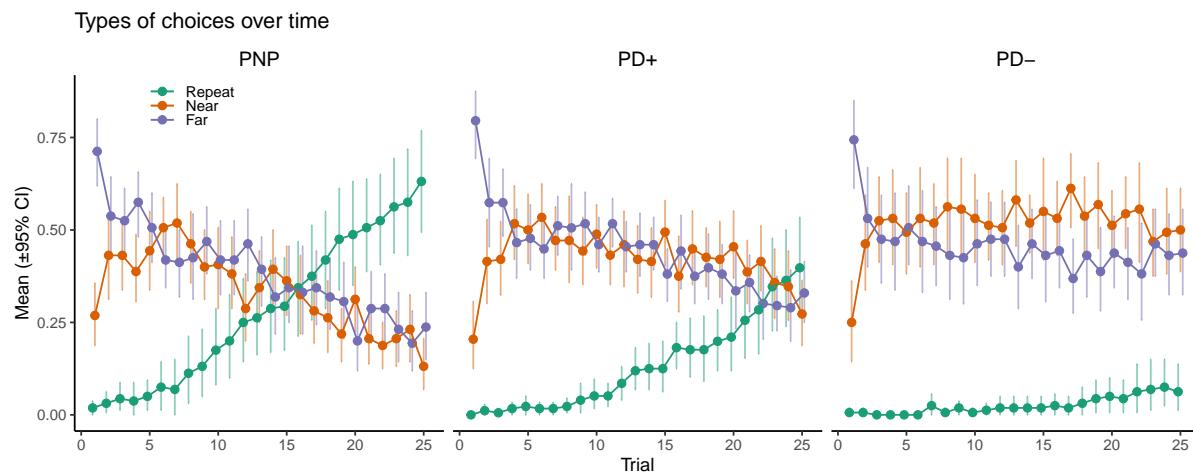
```
df_types_choices_trial_subject <- dat %>%
  filter(round %in% 2:9 & trial > 0) %>%
  group_by(id, group, trial, type_choice) %>%
  summarise(n = n()) %>%
  complete(type_choice, fill = list(n = 0)) %>%  #
  group_by(id, group, trial) %>%
  mutate(prop = n / sum(n)) %>% # Calculate proportion for each type_choice
  ungroup()

ggplot(df_types_choices_trial_subject, aes(x = trial, y = prop, color = type_choice, group =
  facet_wrap(~group) +
  stat_summary(fun.data = mean_cl_boot, geom = "errorbar", width = 0.2, alpha = 0.5, , position = "dodge") +
  stat_summary(fun = mean, geom = "point", size = 2, position=position_dodge(width=0.5)) +
  stat_summary(fun = mean, geom = "line", position=position_dodge(width=0.5)) +
  scale_fill_manual(values = groupcolors) +
  scale_color_manual(values = groupcolors) +
  labs(
```

```

x = "Trial",
y = "Mean (\u00b195% CI)",
title = "Types of choices over time",
color = "Type choice"
) +
theme_classic() +
theme(strip.background = element_blank(),
strip.text = element_text(color = "black", size=12),
legend.position = "inside",
legend.position.inside = c(0.15, 1),    # Use legend.position.inside
legend.justification = c(1, 1),
legend.title = element_blank(),
legend.spacing.y = unit(0.05, 'cm'),
legend.key.height = unit(0.3, 'cm'))
)

```



```
ggsave("plots/types_choice_by_trial.png", dpi=300, width = 8, height = 4)
```

#### 4.6.2 Distance as function of previous reward

Finally, we analysed the relation between the value of a reward obtained at time  $t$  and the search distance on the subsequent trial  $t + 1$ . If a large reward was obtained, searchers should search more locally, while conversely, if a low reward was obtained, searchers should be more likely to search farther away.

First, we examined the correlations based on the raw data, without accounting for the hierarchical structure of the dataset. These analyses revealed a notably stronger correlation for

PNP and PD+ patients compared to PD- patients, suggesting that Parkinson's patients off medication exhibited less adaptive search behavior than those on medication and individuals with polyneuropathies

- PNP:  $r = -.46$ ,  $p < .001$ ,  $BF > 100$
- PD+:  $r = -.34$ ,  $p < .001$ ,  $BF > 100$
- PD-:  $r = -.19$ ,  $p < .001$ ,  $BF > 100$

```
# correlation of previous reward and distance of consecutive choices, by age group and environment
# overall, ignoring within-subject factor
# dat %>%
#   filter(trial != 0 & round %in% 2:9) %>% # exclude first (randomly revealed) tile and practice
#   group_by(group) %>%
#   summarise(corTestPretty(previous_reward, distance))

# mean correlation between distance and reward obtained on previous step
# first aggregated within each round and then within each subject
# such that there is one correlation for each subject

# reward_distance_cor <- dat %>%
#   filter(trial != 0 & round %in% 2:9) %>% # exclude first (randomly revealed) tile and practice
#   group_by(id, round, group) %>%
#   summarise(cor = cor(previous_reward, distance)) %>%
#   mutate(cor = replace_na(cor, 0)) %>% # in some rounds subjects clicked the same tile three times
#   ungroup() %>%
#   group_by(id, group) %>%
#   summarise(mean_cor = mean(cor))

# mean correlation between distance and reward obtained on previous step as function of group
# reward_distance_cor %>%
#   group_by(group) %>%
#   summarise(n = n(),
#             m_cor = mean(mean_cor),
#             SD_cor = sd(mean_cor),
#             se_cor = SD_cor / sqrt(n),
#             lower_ci_cor = m_cor - qt(1 - (0.05 / 2), n - 1) * se_cor,
#             upper_ci_cor = m_cor + qt(1 - (0.05 / 2), n - 1) * se_cor)

#plot regression lines based on raw data
# ggplot(subset(dat, trial > 0 & round %in% 2:9), aes(x = previous_reward, y = distance, color = group))
#   facet_wrap(~group) +
#   geom_jitter(alpha = 0.3, width = 0.1, height = 0.1) +
#   geom_smooth(method = "lm", formula = y ~ x, se = TRUE) +
```

```

# ggtile("Regression Lines for Distance by Previous Reward and Group") +
# theme_minimal() +
# xlab("Previous Reward") +
# ylab("Distance")

```

Given the nested structure of the data, we next employed hierarchical regression analyses to predict search distance based on the reward obtained in the previous step, group, and their interaction as population-level (fixed) effects. We accounted for individual differences by allowing participants to have random intercepts (Note BM: we can also include random slopes when we have more data points, but lmer did not converge and brm also need some tweaks).

These analyses show that both the magnitude of reward obtained on the last step and group influence search distance. Notably, Parkinson patients off medication (PD-) adapted their search behavior less in response to reward magnitude, while patients on medication (PD+) exhibit more adaptation, but still less than the PNP group.

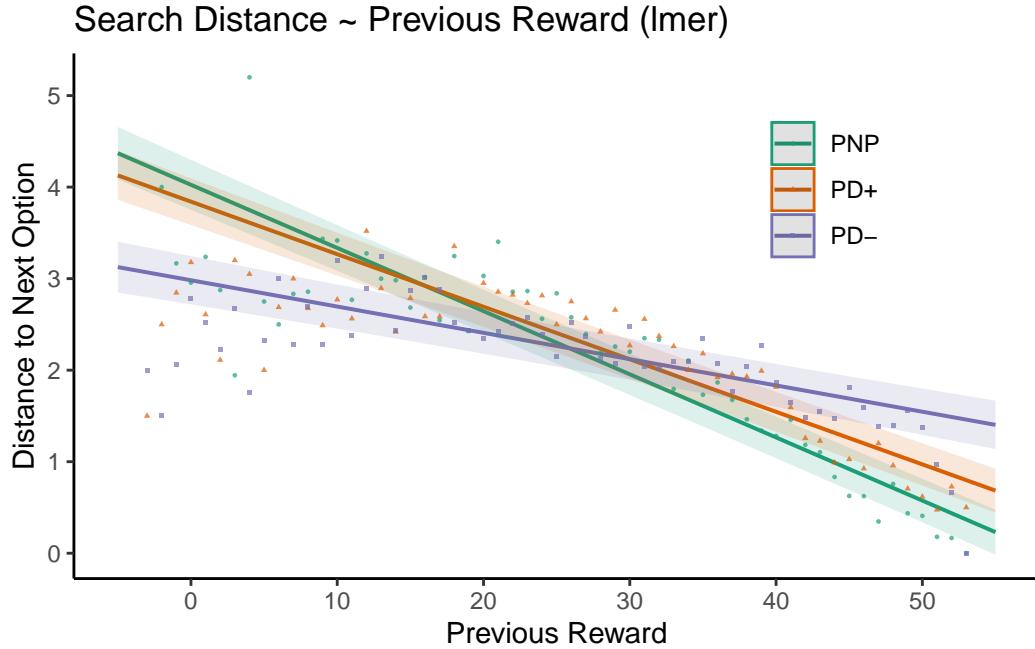
```

# for now, random intercepts only, Random intercept + random slope not stable
# lmer_distance_reward <- lmer(distance ~ previous_reward * group + (previous_reward + group
# data = subset(dat, trial > 0 & round %in% 2:9))
# fit model
lmer_distance_reward <- lmer(distance ~ previous_reward * group + (1 | id),
                               data = subset(dat, trial > 0 & round %in% 2:9))

#summary(lmer_distance_reward)
#emmeans(lmer_distance_reward, pairwise ~ previous_reward | group, pbkrtest.limit = 15000)

plot_model(lmer_distance_reward, type = "pred", terms = c("previous_reward", "group")) +
  stat_summary(dat, mapping=aes(x=previous_reward, y=distance, color=group, fill=group, shape
scale_x_continuous('Previous Reward', breaks = (c(0,10,20,30,40,50))) +
  ylab('Distance to Next Option')+
  scale_fill_manual(values=groupcolors) +
  scale_color_manual(values=groupcolors) +
  ggtile('Search Distance ~ Previous Reward (lmer)') +
  theme_classic() +
  theme(legend.position = "inside",
        legend.position.inside = c(0.85, 0.9),    # Use legend.position.inside
        legend.justification = c(1, 1),
        legend.title = element_blank(),
        legend.box.background = element_blank())

```



```
ggsave("plots/regression_distance_reward_lmer.png", dpi=300, height=3, width=4)
```

Results of a Bayesian hierarchical regression with random intercepts and random slopes for subjects yield comparable findings, again indicating distinct search patterns and responses to reward magnitude across groups, with the PD- group exhibiting less adaptive behavior.

```
# Bayesian regression analysis
# run_model() is a wrapper for brm models such that it saves the full model the first time it
# Fixed effects: previous_reward and group.
# Random effects: random slopes and a random intercept for both previous_reward and group by

# random intercept and random slope
# brm_distance_reward <- run_model(brm(distance ~ previous_reward * group + (previous_reward

# random intercept
brm_distance_reward <- run_model(brm(distance ~ previous_reward * group + (1|id),
                                       data=subset(dat, trial > 0 & round %in% 2:9),
                                       cores=4, # running into problems with cores > 1
                                       seed = 0815,
                                       iter = 5000,
                                       warmup=1000,
                                       control = list(adapt_delta = 0.99, max_treedepth = 15))
#prior = prior(normal(0,10), class = "b")),
```

```

modelName = 'brm_distance_reward')
#tab_model(brm_distance_reward, bpe="mean", title = "Bayesian regression results: Search dist
#bayes_R2(brm_distance_reward)

# Compute marginal effects for distance as a function of previous_reward across groups
# marginal_effects_data <- conditional_effects(brm_distance_reward, effects = "previous_rewa
#
# # Generate conditional effects for previous_reward by group
# effects <- conditional_effects(brm_distance_reward, effects = "previous_reward:group")
#
# # Extract the plot for previous_reward:group
# plot_effect <- plot(effects, plot = FALSE)[[1]]
#
# plot_effect +
#   xlab('Previous Reward')+
#   ylab('Distance to Next Option')+
#   scale_color_viridis(discrete=TRUE, name='Group', direction=1) +
#   scale_fill_viridis(discrete=TRUE, name='Group', direction=1) +
#   ggtitle('Search Distance ~ Previous Reward') +
#   theme_classic() +
#   theme(legend.position = "inside",
#         legend.position.inside = c(0.85, 0.9),    # Use legend.position.inside
#         legend.justification = c(1, 1),
#         legend.title = element_blank())           # Align top-right corner of legend

# generate plot manually predictions (otherwise difficult to plot the mean empirical values
prevReward <- seq(0,50) #/ 50 # normalized reward
group <- levels(dat$group)
newdat <- expand.grid(previous_reward=prevReward, group=group)

# predict distance based on previous reward
preds <- fitted(brm_distance_reward, re_formula=NA, newdata=newdat, probs=c(.025, .975))
predsDF <- data.frame(previous_reward=rep(prevReward, 3),
                      group=rep(levels(dat$group), each=length(prevReward)),
                      distance=preds[,1],
                      lower=preds[,3],
                      upper=preds[,4])

# average distance
grid <- expand.grid(x1=0:7, x2=0:7, y1=0:7, y2=0:7)
grid$distance <- NA

```

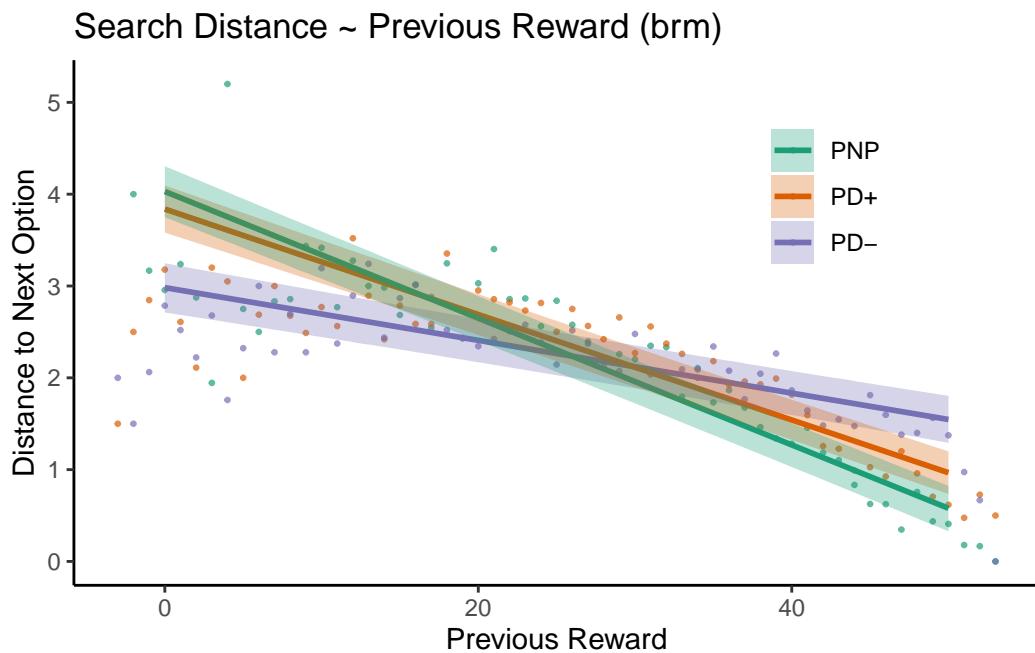
```

for(i in 1:dim(grid)[1]){
  grid$distance[i] <- dist(rbind(c(grid$x1[i], grid$x2[i]), c(grid$y1[i], grid$y2[i])), method="euclidean")
}

meanDist <- mean(grid$distance)

# plot predictions
ggplot() +
  stat_summary(dat, mapping=aes(x=previous_reward, y=distance, color=group, fill=group), fun=mean) +
  geom_line(predsDF, mapping=aes(x=previous_reward, y=distance, color=group), linewidth=1) +
  geom_ribbon(predsDF, mapping=aes(x=previous_reward, y=distance, ymin=lower, ymax=upper, fill=group)) +
  #geom_hline(yintercept=meanDist, linetype='dashed', color='red') + # mean distance
  # xlab('Normalized Previous Reward')+
  xlab('Previous Reward')+
  ylab('Distance to Next Option')+
  scale_fill_manual(values=groupcolors) +
  scale_color_manual(values=groupcolors) +
  ggtitle('Search Distance ~ Previous Reward (brm') +
  theme_classic() +
  theme(legend.position = "inside",
        legend.position.inside = c(0.85, 0.9),
        legend.justification = c(1, 1),
        legend.title = element_blank())

```



```
ggsave("plots/regression_distance_reward_brms.png", dpi=300, height=3, width=4)
```

## 4.7 Bonus round judgments

In the bonus round, participants made 15 search decisions and then predicted the rewards for 5 randomly chosen, previously unobserved tiles. Subsequently, they chose one of the five tiles and continued the round until the search horizon of 25 clicks was met.

Data frame `dat_bonus` contains the following variables:

- `id`: participant id
- `bonus_env_number`: internal id of the bonus round environment
- `bonus_environment`: recodes `condition` as *Smooth* (high spatial correlation)
- `x` and `y` are the coordinates of the random tiles on the grid for which participants were asked to provide reward estimates
- `givenValue`: participant reward judgment (scale 0-50)
- `howSecure`: participant confidence for given reward judgment (scale 0-10)
- `chosen_x` and `chosen_y` are the coordinates of the tile chose after making reward and confidence judgments for 5 random tiles
- `true_z` is the ground truth, i.e. true expected reward of a tile
- `error` is the absolute deviation between participants reward estimates (`givenValue`) and ground truth (`true_z`)
- `chosen` is whether the option was chosen or not (participants chose one of the five options after estimating their value and confidence in their reward prediction)

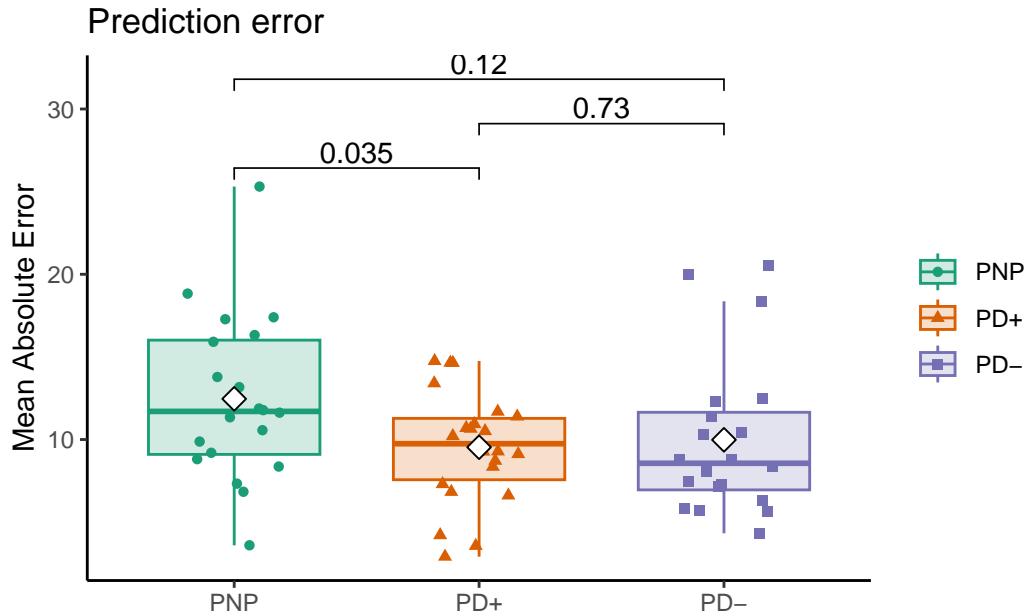
 Note

Charley: is this scaling still correct (taken from YKWG code)? `bonus_environment$z <- bonus_environment$z * scale_factor + 5`

id	bonus_env_number	bonus_environment	x	y	givenValue	howSecure	chosen_x	chosen_y
111	38	Smooth	5	6	20	5	7	3
111	38	Smooth	2	7	26	4	7	3
111	38	Smooth	7	3	16	5	7	3
111	38	Smooth	0	7	28	3	7	3
111	38	Smooth	7	6	30	5	7	3
115	39	Smooth	0	0	19	4	3	1

#### 4.7.1 Prediction error

The plot shows the mean absolute error between participants' estimates and the true underlying expected reward, for each age group and environment.



Compared to a random baseline, all groups performed better than chance level:

- PNP:  $t(19) = -2.2, p = .040, d = 0.5, BF = 1.7$
- PD+:  $t(21) = -7.5, p < .001, d = 1.6, BF > 100$
- PD-:  $t(19) = -4.7, p < .001, d = 1.0, BF > 100$

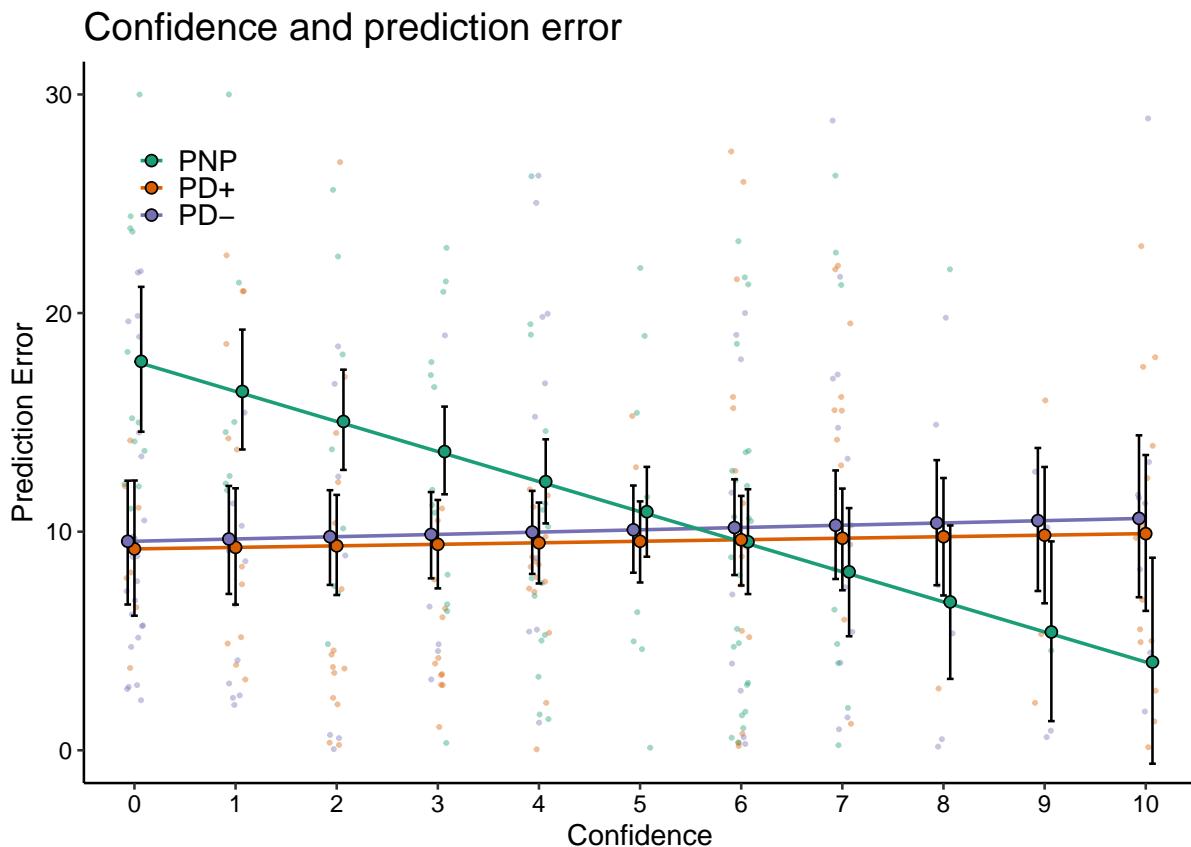
There was no difference between Parkinson patients on and off medication. PNP participants had higher prediction error than Parkinson patients on medication (PD+), but were not worse than Parkinson participants off medication: (PD-).

#### 4.7.2 Prediction error and confidence

```
# Across all judgments and participants, there was no systematic relation between confidence
# corTestPretty(dat_bonus$error, dat_bonus$howSecure, method = "kendall")
# cor.test(dat_bonus$error, dat_bonus$howSecure, method = "kendall")
# correlationBF(dat_bonus$error, dat_bonus$howSecure, method = "kendall")
```

A Bayesian regression with prediction error as dependent variable, and confidence and group and their interaction as population-level (“fixed”) effects, and a random intercept for participants showed that for PNP patients confidence and prediction error were negatively correlated (i.e., lower confidence was associated with a higher error), whereas for the two Parkinson groups there was no relation.

```
brm_bonus_confidence_error_by_group <- run_model(brm(error ~ howSecure * group + (1|id),
  data=dat_bonus,
  cores=4,
  control = list(adapt_delta = 0.99),
  seed = 0815),
  modelName = 'brm_bonus_confidence_error_by_gro'
#tab_model(brm_bonus_confidence_error_by_group, bpe = "mean", title = "Bayesian regression re
#bayes_R2(brm_bonus_confidence_error_by_group)
```



#### 4.7.3 Analysis of selected tiles

To analyze selected and not-selected options, we first averaged the predicted reward and confidence of the not-chosen tiles within subjects, and then compared chosen and not chosen options. Selected tiles tended to have higher predicted rewards. Participants were not more confident in selected options, and selected tiles did not have a higher true reward than not selected tiles.

```
# average not-chosen tiles within subjects first
df_chosen_overall <- dat_bonus %>%
  group_by(id, chosen) %>%
  summarise(m_givenValue = mean(givenValue),
            m_howSecure = mean(howSecure),
            m_true_z = mean(true_z))

# df_chosen_overall %>%
#   group_by(chosen) %>%
#   summarise(predicted_reward = mean(m_givenValue),
#             confidence = mean(m_howSecure),
#             true_reward = mean(m_true_z)) %>%
#   kable(format = "html", escape = F, digits = 2) %>%
#   kable_styling("striped", full_width = F)

df_chosen_group <- dat_bonus %>%
  group_by(id, group, chosen) %>%
  summarise(m_givenValue = mean(givenValue),
            m_howSecure = mean(howSecure),
            m_true_z = mean(true_z))

# df_chosen_group %>%
#   group_by(group, chosen) %>%
#   summarise(predicted_reward = mean(m_givenValue),
#             confidence = mean(m_howSecure),
#             true_reward = mean(m_true_z)) %>%
#   kable(format = "html", escape = F, digits = 2) %>%
#   kable_styling("striped", full_width = F)

# chosen vs not chosen: predicted reward

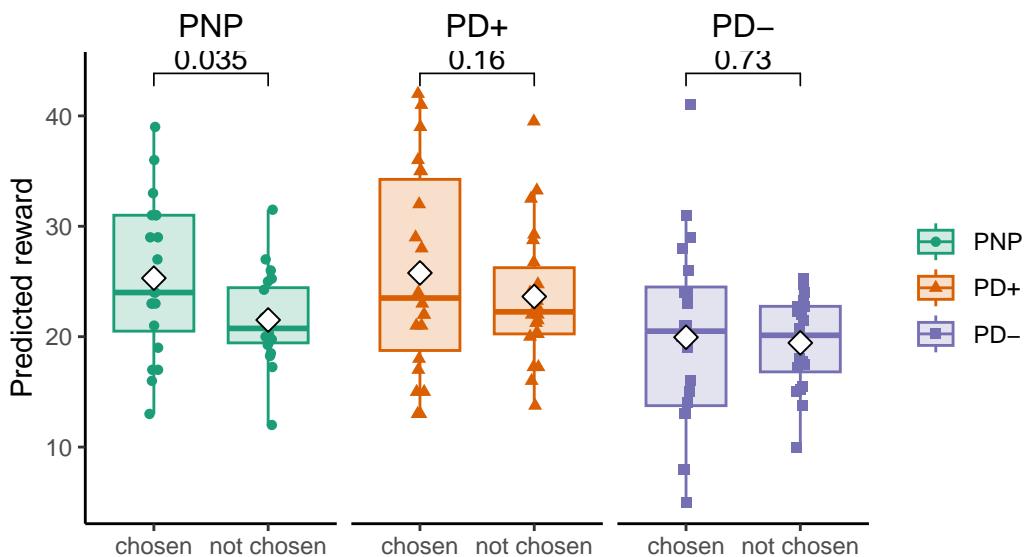
ggboxplot(df_chosen_group, x = "chosen", y = "m_givenValue",
          color = "group", palette = groupcolors, fill = "group", alpha = 0.2,
          add = "jitter", shape = "group", title = "Predicted reward of chosen vs. not chosen",
          facet.by = "group") +
```

```

    ylab("Predicted reward") +
    xlab("") +
    stat_compare_means(comparisons = list( c("chosen", "not chosen") ), paired = T, method = "t",
    stat_summary(fun = mean, geom="point", shape = 23, fill = "white", size=3) +
    theme_classic() +
    theme(strip.background = element_blank(),
        strip.text = element_text(color = "black", size=12),
        legend.title = element_blank()
    )
)

```

Predicted reward of chosen vs. not chosen options



```

ggsave("plots/bonusround_chosen_not_chosen_options_predicted_reward.png", dpi=300, width=7, height=5)

# chosen vs not chosen: true reward

ggboxplot(df_chosen_group, x = "chosen", y = "m_true_z",
           color = "group", palette = groupcolors, fill = "group", alpha = 0.2,
           add = "jitter", shape = "group", title = "True reward of chosen vs. not chosen options",
           facet.by = "group") +
    ylab("True reward") +
    xlab("") +
    stat_compare_means(comparisons = list( c("chosen", "not chosen") ), paired = T, method = "t",
    stat_summary(fun = mean, geom="point", shape = 23, fill = "white", size=3) +
    theme_classic() +
    theme(strip.background = element_blank(),
        strip.text = element_text(color = "black", size=12),
        legend.title = element_blank()
    )
)

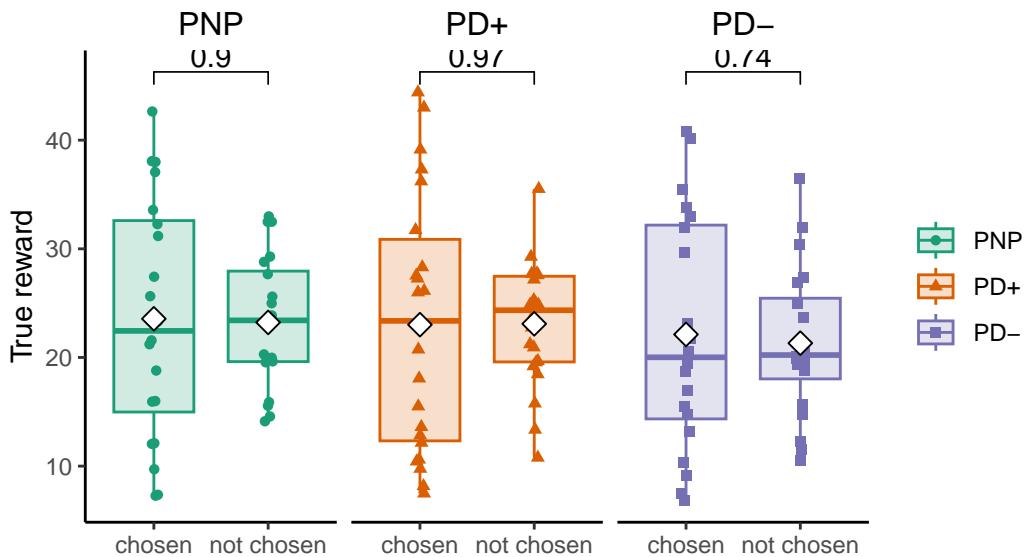
```

```

        strip.text = element_text(color = "black", size=12),
        legend.title = element_blank()
    )

```

True reward of chosen vs. not chosen options



```

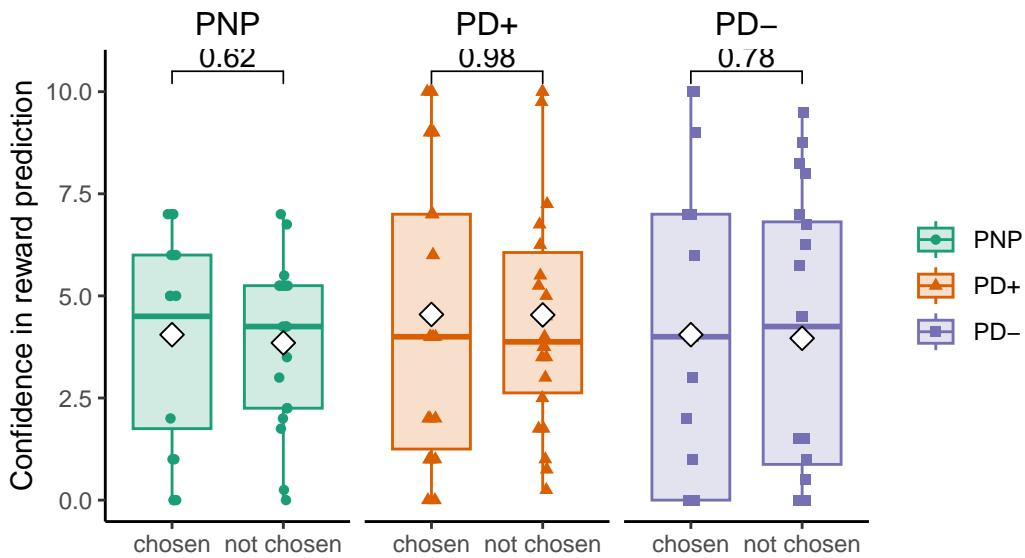
ggsave("plots/bonusround_chosen_not_chosen_options_predicted_reward.png", dpi=300, width=7, height=5)

# chosen vs not chosen: confidence

ggbboxplot(df_chosen_group, x = "chosen", y = "m_howSecure",
            color = "group", palette = groupcolors, fill = "group", alpha = 0.2,
            add = "jitter", shape = "group", title = "Confidence of chosen vs. not chosen options",
            facet.by = "group") +
  ylab("Confidence in reward prediction") +
  xlab("") +
  stat_compare_means(comparisons = list( c("chosen", "not chosen") ), paired = T, method = "t.test",
                     p.adjust.method = "bonferroni") +
  stat_summary(fun = mean, geom="point", shape = 23, fill = "white", size=3) +
  theme_classic() +
  theme(strip.background = element_blank(),
        strip.text = element_text(color = "black", size=12),
        legend.title = element_blank())
)

```

### Confidence of chosen vs. not chosen options



```

ggsave("plots/bonusround_chosen_not_chosen_options_confidence.png", dpi=300, width=7, height=5)

# ggplot(df_mean_reward_subject, aes(x=group, y=mean_reward, group = group, fill = group)) +
#   #geom_hline(data=filter(df_random_performance, environment=="Rough"), linetype="dotted",
#   #   geom_boxplot(outlier.shape = NA, width = w_box) +
#   stat_summary(fun = mean, geom="point", shape = 23, fill = "white", size=1.8) +
#   geom_jitter(aes(x = as.numeric(group) + 0.1 + 0.15, colour=group), shape = 21, size =
#   scale_fill_manual(values=groupcolors) +
#   #scale_color_viridis(discrete=TRUE, name='Group', direction=1) +
#   #scale_fill_viridis(discrete=TRUE, name='Group', direction=1) +
#   scale_y_continuous("Mean reward", limits = c(18, 42), breaks = seq(20,40,5)) +
#   scale_x_discrete("") +
#   ggttitle("Performance of groups") +
#   theme_classic() +
#   theme(#aspect.ratio = 1,
#         plot.title = element_text(hjust = 0.5, size=10),
#         legend.title = element_blank(),
#         legend.position = 'none',
#         legend.text = element_text(colour="black"),
#         strip.background = element_blank(),
#         axis.text = element_text(colour = "black"),
#         panel.grid.major = element_blank(),
#         panel.grid.minor = element_blank())

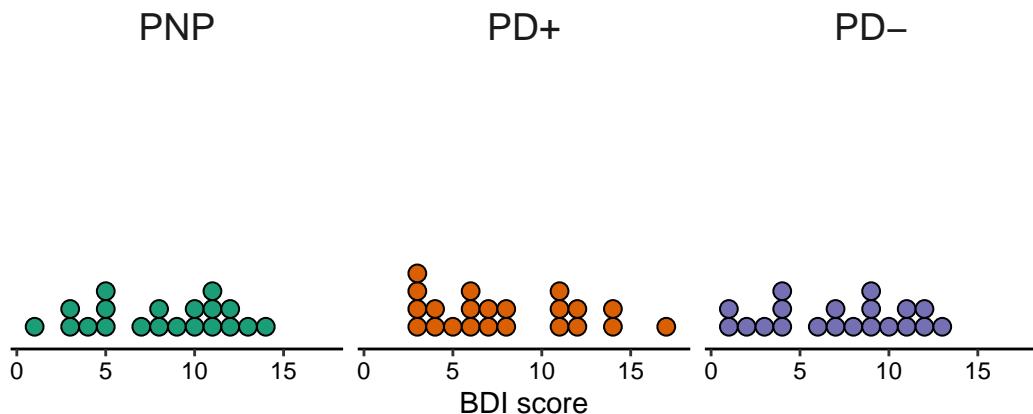
```

```
#  
# ggsave("plots/performance_by_group.png", dpi=300, height = 3, width = 5 )
```

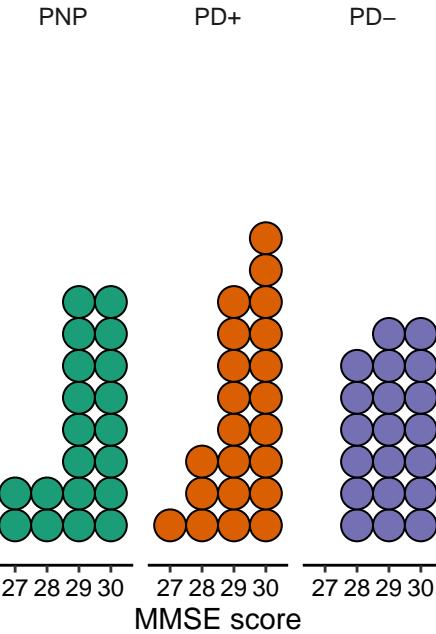
# 5 Appendix

## 5.1 Distribution of BDI, MMSE, and YH in each group

```
# dotplot BDI
# p_dotplot_BDI <-
ggplot(df_sample, aes(x = BDI, fill = group)) +
  facet_wrap(~group) +
  geom_dotplot(binwidth = 1, dotsize = 1) +
  scale_fill_manual(values = groupcolors) +
  scale_x_continuous("BDI score") +
  scale_y_continuous(NULL, breaks = NULL) +
  coord_fixed(ratio = 15) +
  theme_classic() +
  theme(
    legend.title = element_blank(),
    legend.position = 'none',
    strip.text = element_text(size=14),
    legend.text = element_text(colour="black"),
    text = element_text(colour = "black"),
    strip.background =element_blank(),
    axis.text.x = element_text(colour="black"),
    axis.text.y = element_text(colour="black"),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank())
```



```
# p_dotplot_MMSE <-
ggplot(df_sample, aes(x = MMSE, fill = group)) +
  facet_wrap(~group) +
  geom_dotplot(binwidth = 1, dotsize = 1) +
  scale_fill_manual(values = groupcolors) +
  scale_x_continuous("MMSE score") +
  scale_y_continuous(NULL, breaks = NULL) +
  coord_fixed(ratio = 15) +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5, size = 10),
        legend.title = element_blank(),
        legend.position = 'none',
        legend.text = element_text(colour="black"),
        text = element_text(colour = "black"),
        strip.background = element_blank(),
        axis.text.x = element_text(colour="black"),
        axis.text.y = element_text(colour="black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```

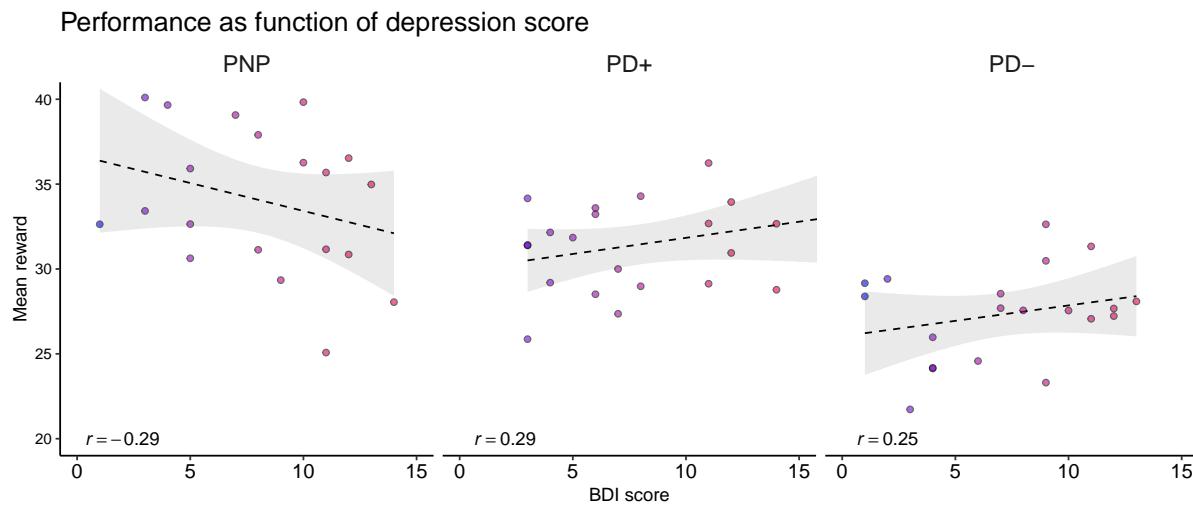


```
p_dotplot_HY <- ggplot(filter(df_sample, group != "PNP"), aes(x = hoehn_yahr, fill = group)) +
  facet_wrap(~group) +
  geom_dotplot(binwidth = 1, dotsize = 1) +
  scale_fill_manual(values = groupcolors) +
  scale_x_continuous("Hoehn-Yahr score") +
  scale_y_continuous(NULL, breaks = NULL) +
  coord_fixed(ratio = 15) +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5, size = 10),
        legend.title = element_blank(),
        legend.position = 'none',
        legend.text = element_text(colour="black"),
        text = element_text(colour = "black"),
        strip.background = element_blank(),
        axis.text.x = element_text(colour="black"),
        axis.text.y = element_text(colour="black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())
```

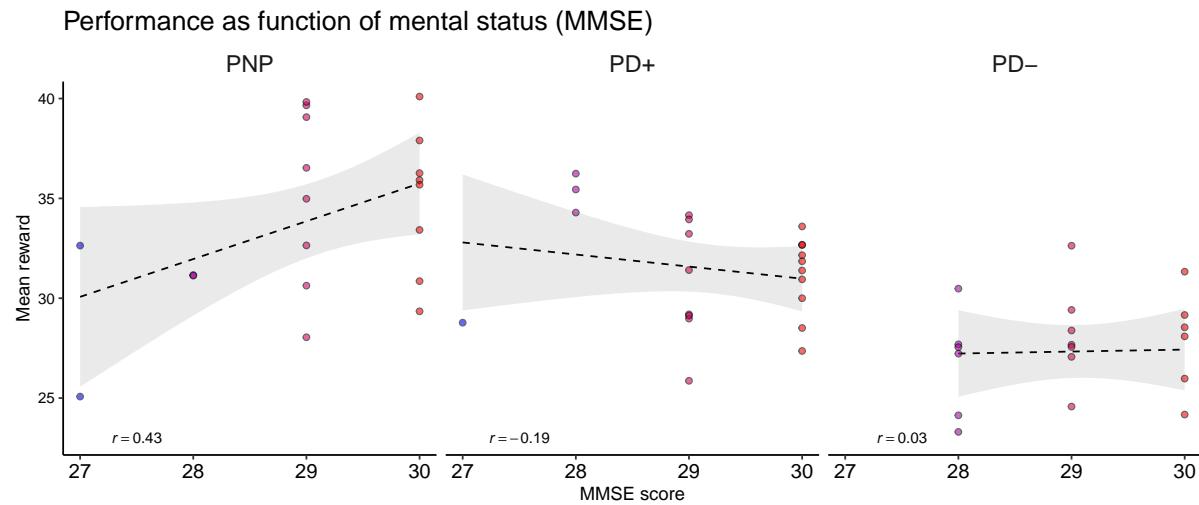
## 5.2 Performance as function of BDI, MMSE, and Hoehn-Yahr

### 5.2.1 Performance as function of depression score (BDI)

The plots show performance as function of depression score (BDI-II), separately for each group. Opposing trends were found in the different groups: for patients with polyneuropathy (PNP), there was a negative relation such that patients with higher depression scores obtained *lower* rewards. For the two Parkinson groups, the relation was positive, such that patients reporting more severe symptoms obtained *higher* rewards.



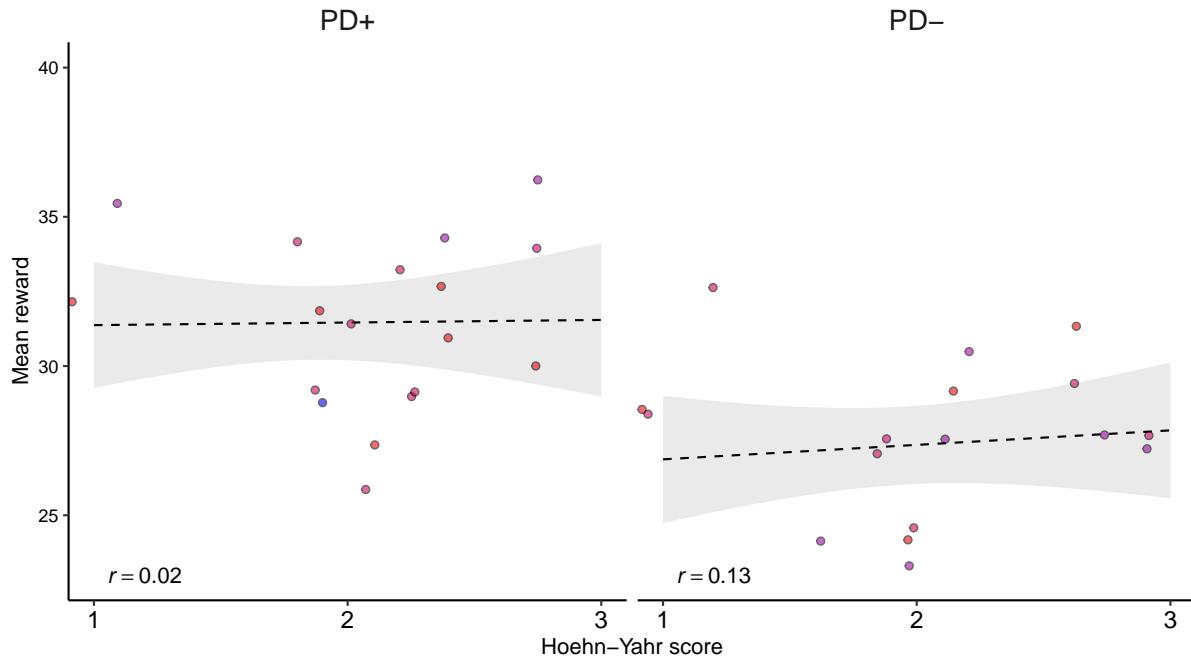
### 5.2.2 Performance as function of Mini-Mental State Examination (MMSE)



### 5.2.3 Performance as function of Hoehn-Yahr (Parkinson patients only)

The Hoehn-Yahr scale provides information about the severity of motor impairments in Parkinson's disease, with higher scores indicating greater severity.

## Performance as function of Hoehn–Yahr score



- Abbott, A. (2010). Levodopa: The story so far. *Nature*, 466(7310), S6–S7.
- Beck, A. T., Steer, R. A., Brown, G. K., et al. (1996). *Beck depression inventory*.
- Folstein, M. F., Folstein, S. E., & McHugh, P. R. (1975). “Mini-mental state”: A practical method for grading the cognitive state of patients for the clinician. *Journal of Psychiatric Research*, 12(3), 189–198.
- Giron, A. P., Ciranka, S., Schulz, E., Bos, W. van den, Ruggeri, A., Meder, B., & Wu, C. M. (2023). Developmental changes in exploration resemble stochastic optimization. *Nature Human Behaviour*, 7(11), 1955–1967.
- Goetz, C. G., Poewe, W., Rascol, O., Sampaio, C., Stebbins, G. T., Counsell, C., Giladi, N., Holloway, R. G., Moore, C. G., Wenning, G. K., et al. (2004). Movement disorder society task force report on the hoehn and yahr staging scale: Status and recommendations the movement disorder society task force on rating scales for parkinson’s disease. *Movement Disorders*, 19(9), 1020–1028.
- Hautzinger, M., Keller, F., & Kühner, C. (2006). *Beck depressions-inventar (BDI-II)*. Harcourt Test Services.
- Hoehn, M. M., & Yahr, M. D. (1967). Parkinsonism: Onset, progression, and mortality. *Neurology*, 17(5), 427–427.
- Meder, B., Wu, C. M., Schulz, E., & Ruggeri, A. (2021). Development of directed and random exploration in children. *Developmental Science*, 24(4), e13095. <https://doi.org/https://doi.org/10.1111/desc.13095>
- Sadeghiyeh, H., Wang, S., Alberhasky, M. R., Kyllo, H. M., Shenhav, A., & Wilson, R. C. (2020). Temporal discounting correlates with directed exploration but not with random

- exploration. *Scientific Reports*, 10(1), 4020.
- Schulz, E., Wu, C. M., Ruggeri, A., & Meder, B. (2019). Searching for rewards like a child means less generalization and more directed exploration. *Psychological Science*, 30(11), 1561–1572. <https://doi.org/10.1177/0956797619863663>
- Tambasco, N., Romoli, M., & Calabresi, P. (2018). Levodopa in parkinson's disease: Current status and future developments. *Current Neuropharmacology*, 16(8), 1239–1252.
- Wu, C. M., Schulz, E., Speekenbrink, M., Nelson, J. D., & Meder, B. (2018). Generalization guides human exploration in vast decision spaces. *Nature Human Behaviour*, 2, 915–924. <https://doi.org/10.1038/s41562-018-0467-4>