Analysis of modeling results

Analysis of modeling results for the paper “Acquiring Constraints on Filler-Gap Dependencies with Structural Collocations: Assessing a Computational Learning Model of Island-Insensitivity in Norwegian” (submitted to *Language Acquisition*).

Loading the required packages and the data

```
rm(list = ls())  # removing everything from the environment
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.2.1     v dplyr   1.1.4
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

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

```
library(ggpubr)
library(readxl)
library(grid)
```

Loading in the data: modeling, experimental, and bootstrapped data (for CI)

```
# Experimental data
exp_data = read_excel("data/cond_means_kobzeva_et_al_2022.xlsx")

# Data for graph showing island/no island effect (the Sprouse design)
explanation_graph = read_excel("data/explanation_graph.xlsx")

# Modeling data
model_data = read_excel("data/model_results_all.xlsx")
```

```
# Bootstrapped data was simulated separately for each n-gram/dependency combo
boot_rc2 = read.csv("data/bootstrap_results_rc_bigrams.csv")
boot_rc3 = read.csv("data/bootstrap_results_rc_trigrams.csv")
boot_wh2 = read.csv("data/bootstrap_results_wh_bigrams.csv")
boot_wh3 = read.csv("data/bootstrap_results_wh_trigrams.csv")
```

```
# Merging it all together
boot_data = bind_rows(boot_rc2, boot_rc3, boot_wh2, boot_wh3)
```

```
# Calculating the bootstrapped CIs for modeling data
boot_avg = boot_data %>%
  arrange(raw_probability) %>% # ordering the data
  group_by(distance, structure, condition, dependency, n_gram) %>%
  slice(26:975) %>% # takes 950 rows in the middle
  summarise(
    n = n(), # to make sure it's the right number of observations
    upper = max(log(raw_probability)), # upper bound of 95% CI
    lower = min(log(raw_probability)), # lower bound of 95% CI
    avg_log_prob = log(mean(raw_probability))
            )
```

```
## `summarise()` has grouped output by 'distance', 'structure', 'condition',
## 'dependency'. You can override using the `.groups` argument.
```

```
# Adding bootstrapped CIs to modeling data
model_data = merge(model_data, boot_avg)
```

Plotting the results

```
# Before plotting, ensure that variables are handled correctly
# Making sure that these variables are handled as factors
factors <- c("distance", "structure", "condition", "dependency", "n_gram")
model_data[factors] <- lapply(model_data[factors], factor)

# Probability is a numeric variable
model_data$log_probability = as.numeric(model_data$log_probability)

# So are z-scores, errors, and the number of participants
exp_data$zscores = as.numeric(exp_data$zscores)
exp_data$error = as.numeric(exp_data$error)
exp_data$nn = as.numeric(exp_data$nn)
```

```
# These are four main experiments reported in the paper
exp_subj = subset(exp_data, condition=="Subject")
exp_adj = subset(exp_data, condition=="Adjunct")
exp_eq = subset(exp_data, condition=="EQ")
exp_rc = subset(exp_data, condition=="RC")

model_subj = subset(model_data, condition=="Subject")
model_eq = subset(model_data, condition=="EQ")
model_adj = subset(model_data, condition=="Adjunct")
model_rc = subset(model_data, condition=="RC-predlink")
```

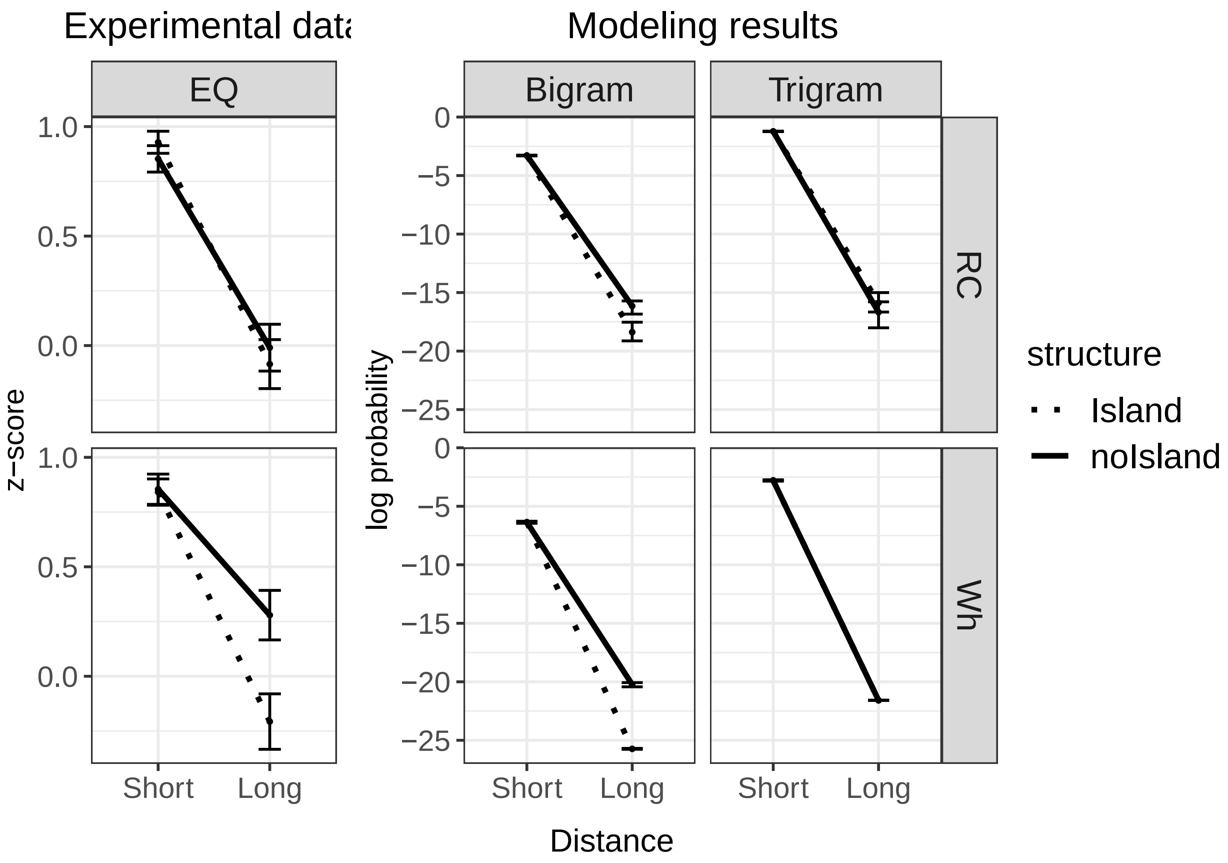
```
# Function for plotting exp behavioral data (z-scores)
plot_zscore = function(data){
  plot =
    ggplot(data, aes(x=factor(ordered(distance, levels=c("Short","Long"))), y=zscores)) +
  geom_point(data=data, aes(y=zscores), size=.5) +
    geom_errorbar(data=data, aes(y=zscores, ymin=zscores-1.96*error, ymax=zscores+1.96*error),
                  width = .2, position=position_dodge(width = 0.9)) +
    geom_line(data=data, aes(y=zscores, group=structure,
                             linetype=structure), linewidth=1) +
  scale_linetype_manual(values=c("dotted", "solid")) +
    xlab("Distance") + ylab("z-score") + facet_grid(dependency~condition) +
  ggtitle("Experimental data") +
  # ylim(-1.1, 1.1) +
  theme_bw() + theme(axis.text=element_text(size = 11),
                     axis.title=element_text(size = 11)) +
```

```r
  theme(plot.title = element_text(hjust = 0.5, size = 14)) +  # title label
  theme(legend.text = element_text(size = 13), legend.title = element_text(size = 13)) +
  theme(strip.text = element_text(size = 13)) + # facet label
  theme(strip.text.y = element_blank())

  return(plot)
}
```

```r
# Function for plotting modeling data (probability)
plot_prob = function(data){
  plot =
    ggplot(data, aes(x=factor(ordered(distance, levels=c("Short","Long"))), y=log_probability)) +
  geom_point(data=data, aes(y=log_probability), size=.5) +
    geom_line(data=data, aes(y=log_probability, group=structure,
                             linetype=structure), linewidth=1) +
  scale_linetype_manual(values=c("dotted", "solid")) +
  geom_errorbar(data=data, aes(y=log_probability, ymin=lower, ymax=upper),
                  width = .2, position=position_dodge(width = 0.9)) +
  ggtitle("Modeling results") +
  # ylim(-39,0) +
    xlab("Distance") + ylab("log probability") + facet_grid(dependency~n_gram) +
  theme_bw() + theme(axis.text=element_text(size = 11),
                     axis.title=element_text(size = 11)) +
  theme(plot.title = element_text(hjust = 0.5, size = 14)) +  # title label
  theme(legend.text = element_text(size = 13), legend.title = element_text(size = 13)) +
  theme(strip.text = element_text(size = 13))  # facet label

  return(plot)
}
```

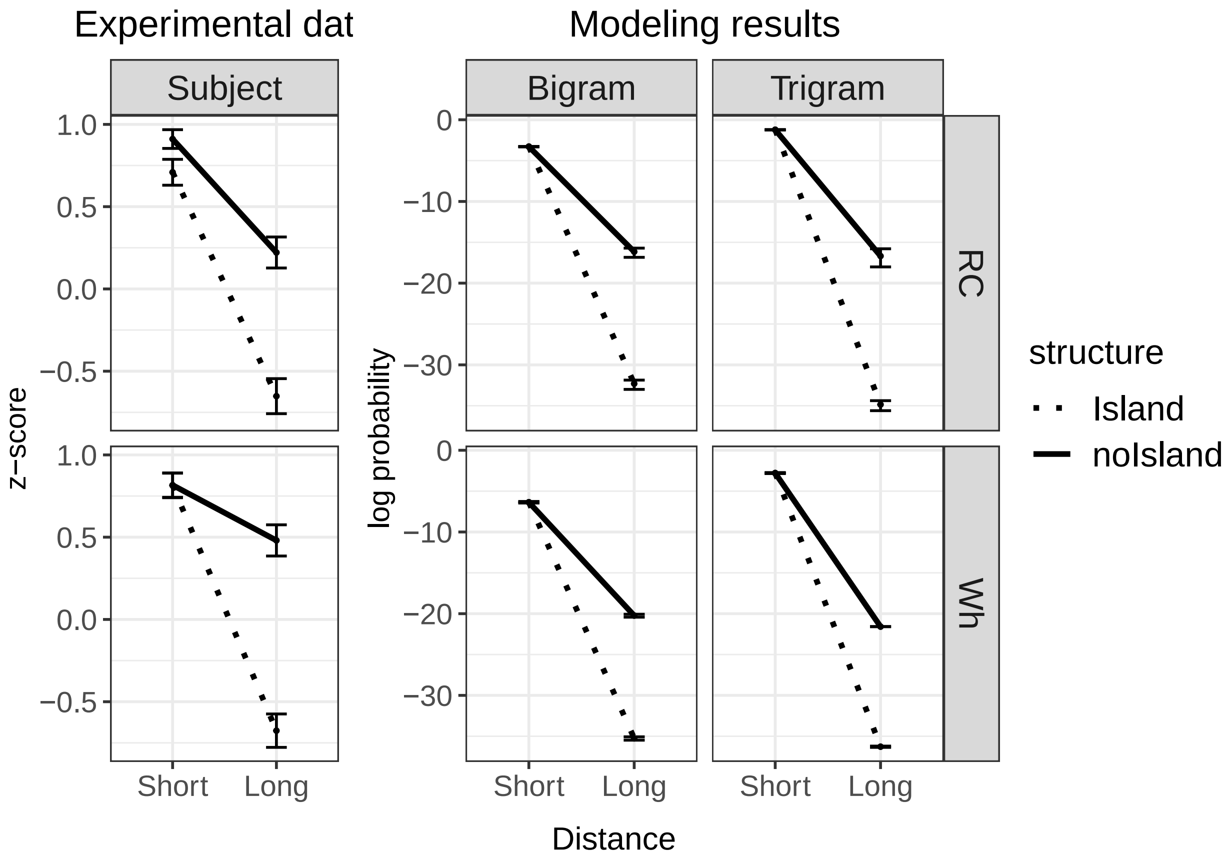
```r
m_eq = plot_prob(model_eq)
e_eq = plot_zscore(exp_eq)
```

```r
plot_eq = ggarrange(e_eq + rremove("xlab"), m_eq + rremove("xlab"), ncol=2, common.legend = TRUE,
                    legend = "right", widths = c(0.55, 1))
require(grid)   # for the textGrob() function
annotate_figure(plot_eq, bottom = textGrob("Distance", gp = gpar(cex = 1)))
```



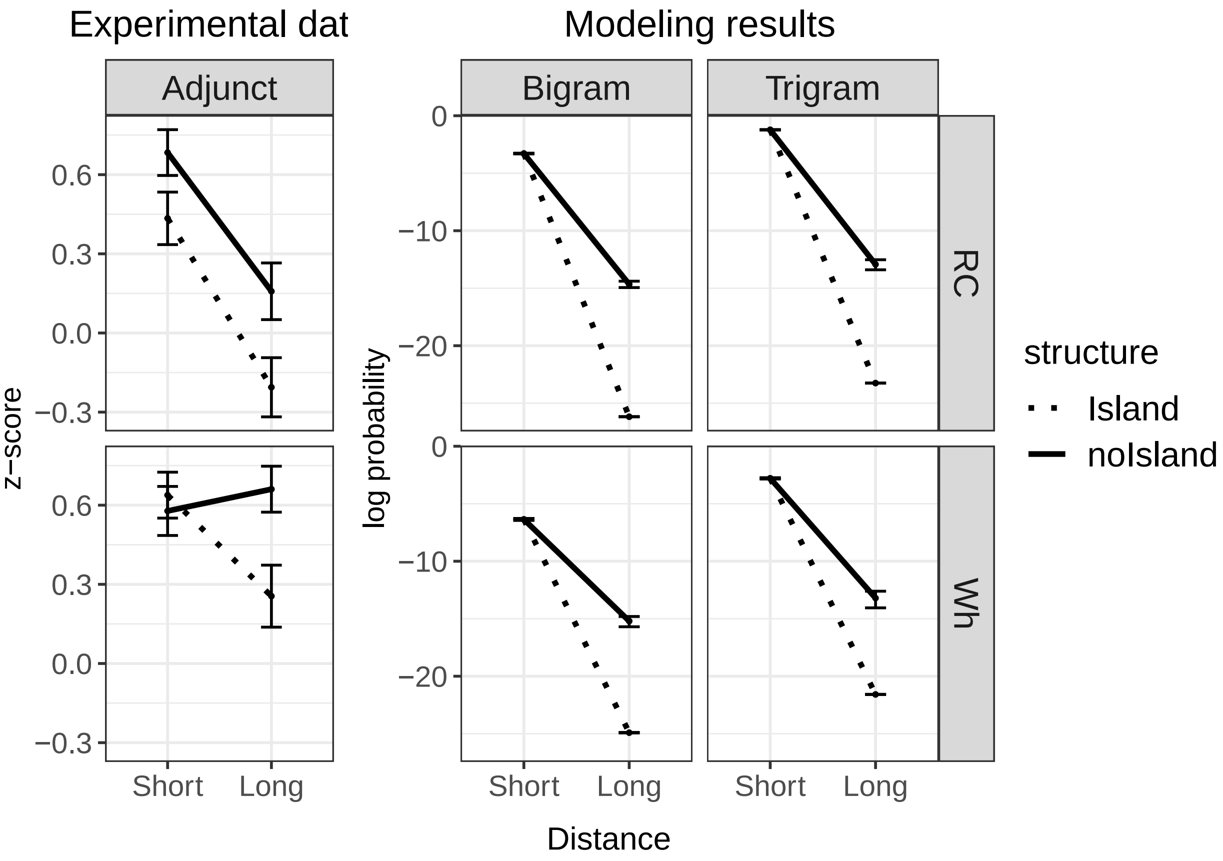
```r
ggsave("plots/eq_common.pdf", width = 10, height = 5)
```

```r
m_subj = plot_prob(model_subj)
e_subj = plot_zscore(exp_subj)

plot_subj = ggarrange(e_subj + rremove("xlab"), m_subj + rremove("xlab"), ncol=2, common.legend = TRUE,
                    legend = "right", widths = c(0.55, 1))
require(grid)   # for the textGrob() function
annotate_figure(plot_subj, bottom = textGrob("Distance", gp = gpar(cex = 1)))
```



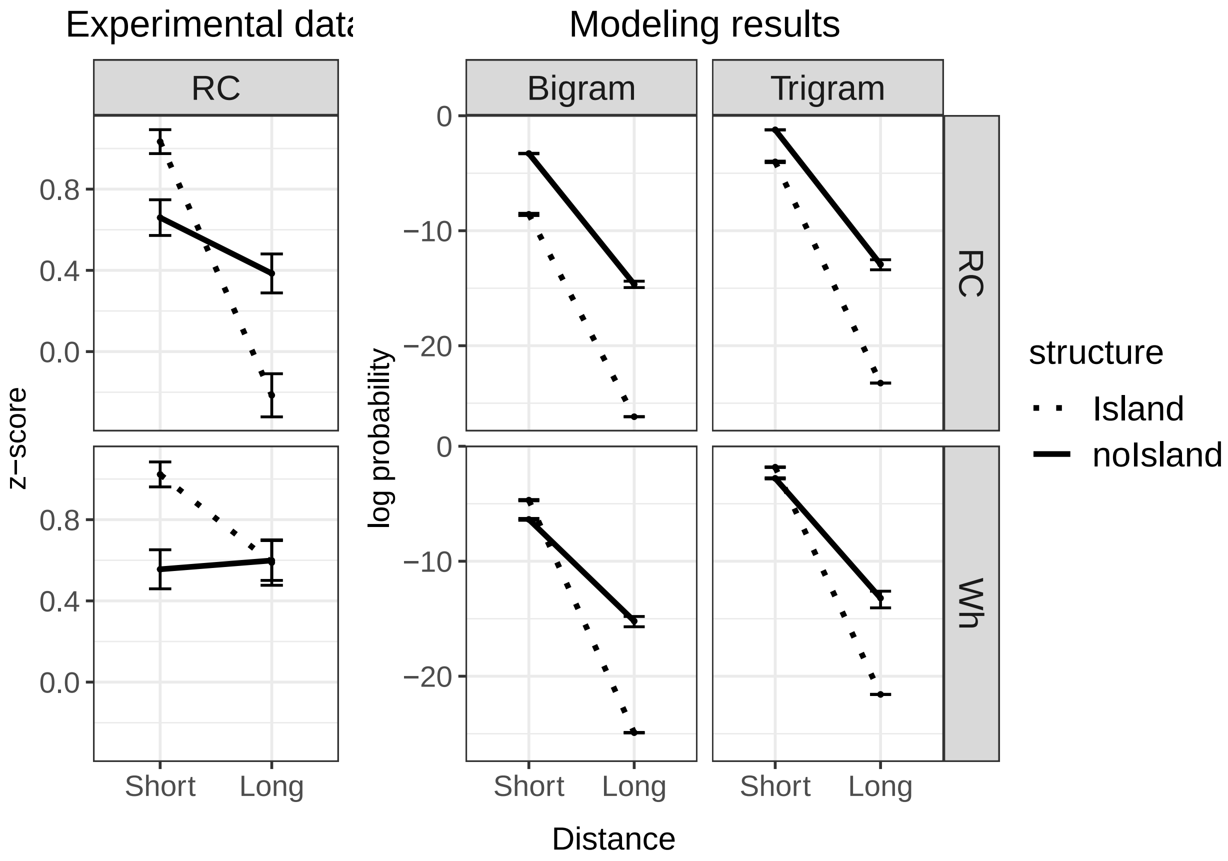
```
ggsave("plots/subj_common.pdf", width = 10, height = 5)
```

```
m_adj = plot_prob(model_adj)
e_adj = plot_zscore(exp_adj)

plot_adj = ggarrange(e_adj + rremove("xlab"), m_adj + rremove("xlab"), ncol=2, common.legend = TRUE,
                    legend = "right", widths = c(0.55, 1))
require(grid)    # for the textGrob() function
annotate_figure(plot_adj, bottom = textGrob("Distance", gp = gpar(cex = 1)))
```



```r
ggsave("plots/adj_common.pdf", width = 10, height = 5)
```

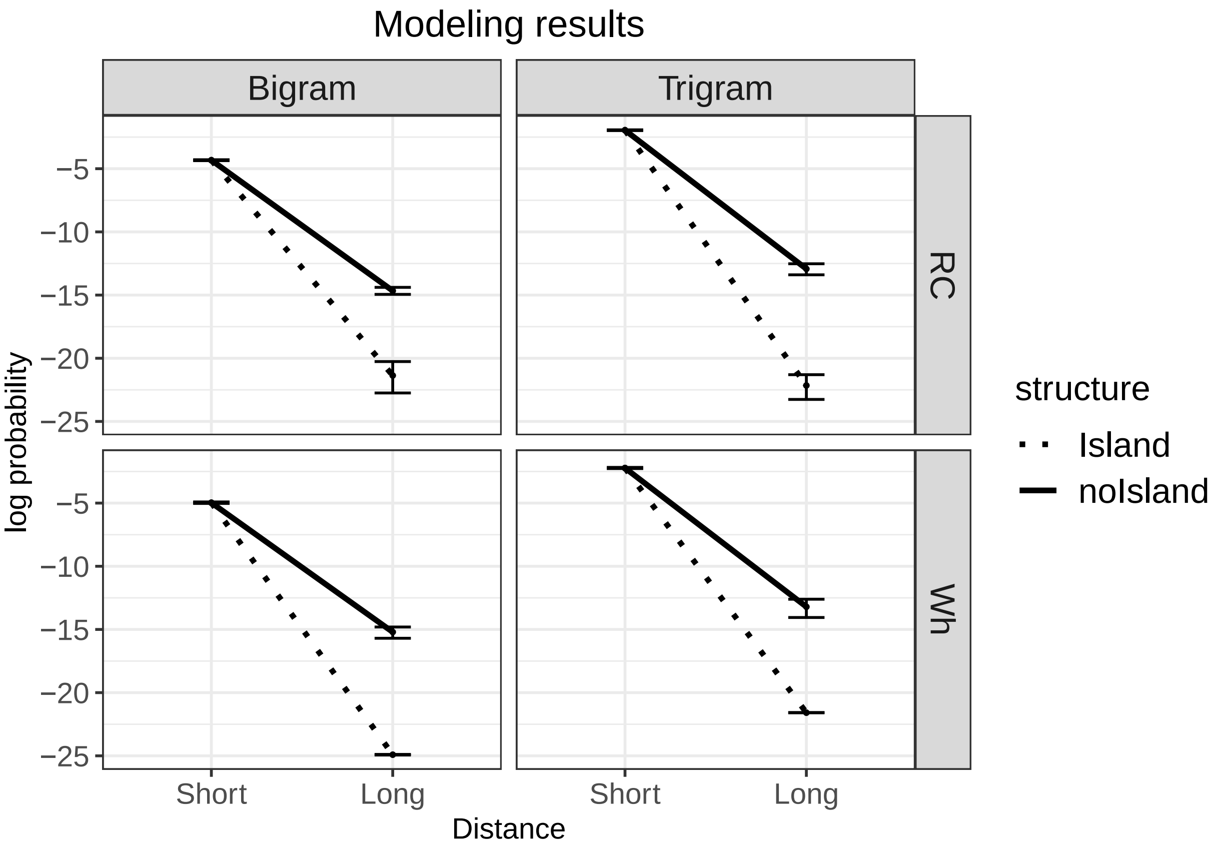
```r
m_rc = plot_prob(model_rc)
e_rc = plot_zscore(exp_rc)

plot_rc = ggarrange(e_rc + rremove("xlab"), m_rc + rremove("xlab"), ncol=2, common.legend = TRUE,
                    legend = "right", widths = c(0.55, 1))
require(grid)   # for the textGrob() function
annotate_figure(plot_rc, bottom = textGrob("Distance", gp = gpar(cex = 1)))
```



```r
ggsave("plots/rc_common.pdf", width = 10, height = 5)
```

```r
# Additional comparisons for the Appendix
model_cnp = subset(model_data, condition=="CNP")
model_eq_obj = subset(model_data, condition=="EQ-object")
model_whether_subj = subset(model_data, condition=="Whether-subject")
model_whether_obj = subset(model_data, condition=="Whether-object")
model_rc_subj = subset(model_data, condition=="RC-subject")
model_rc_pcomp = subset(model_data, condition=="RC-Pcomp")
```

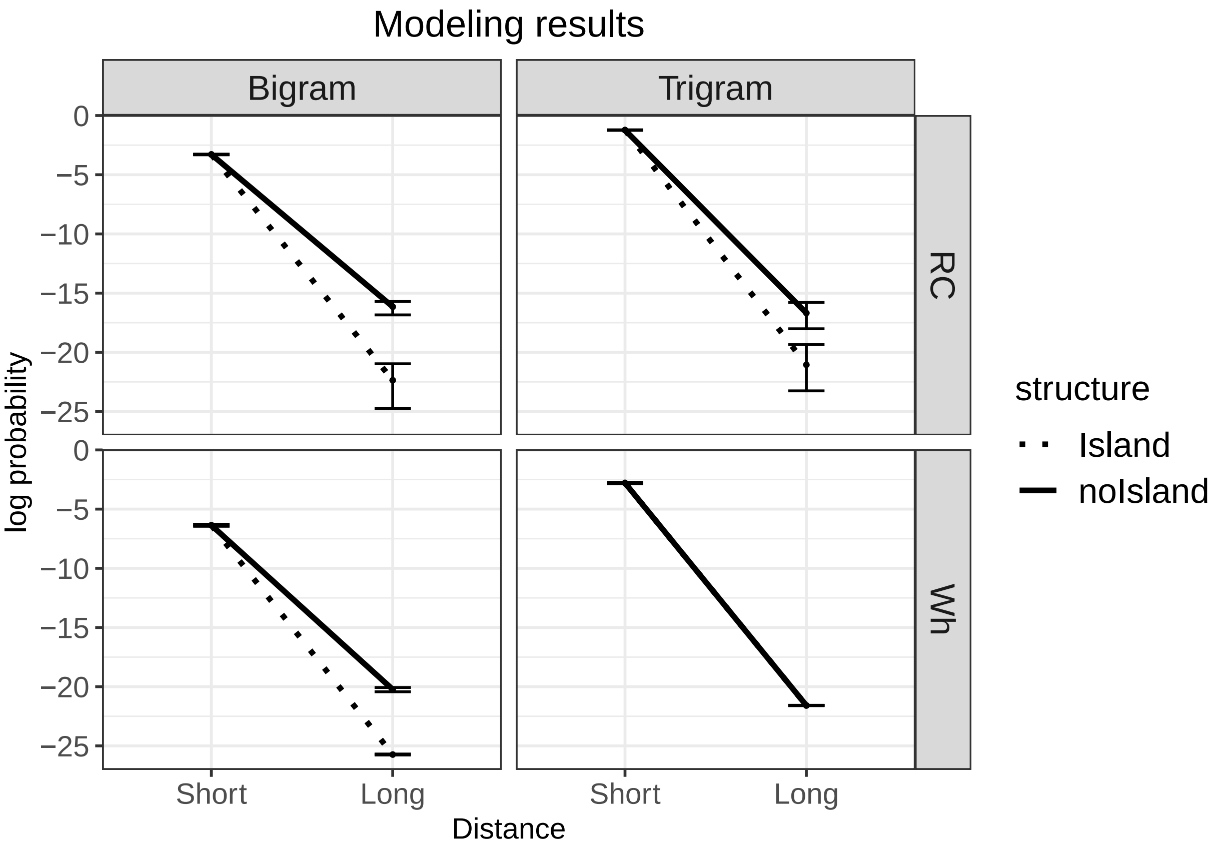
```r
plot_prob(model_eq_obj)
```



```
ggsave("plots/apdx_eq_obj.pdf", height = 5)
```

```
## Saving 6.5 x 5 in image
```

```
plot_prob(model_whether_subj)
```



```
ggsave("plots/apdx_whether_subj.pdf", height = 5)
```

```
## Saving 6.5 x 5 in image
```

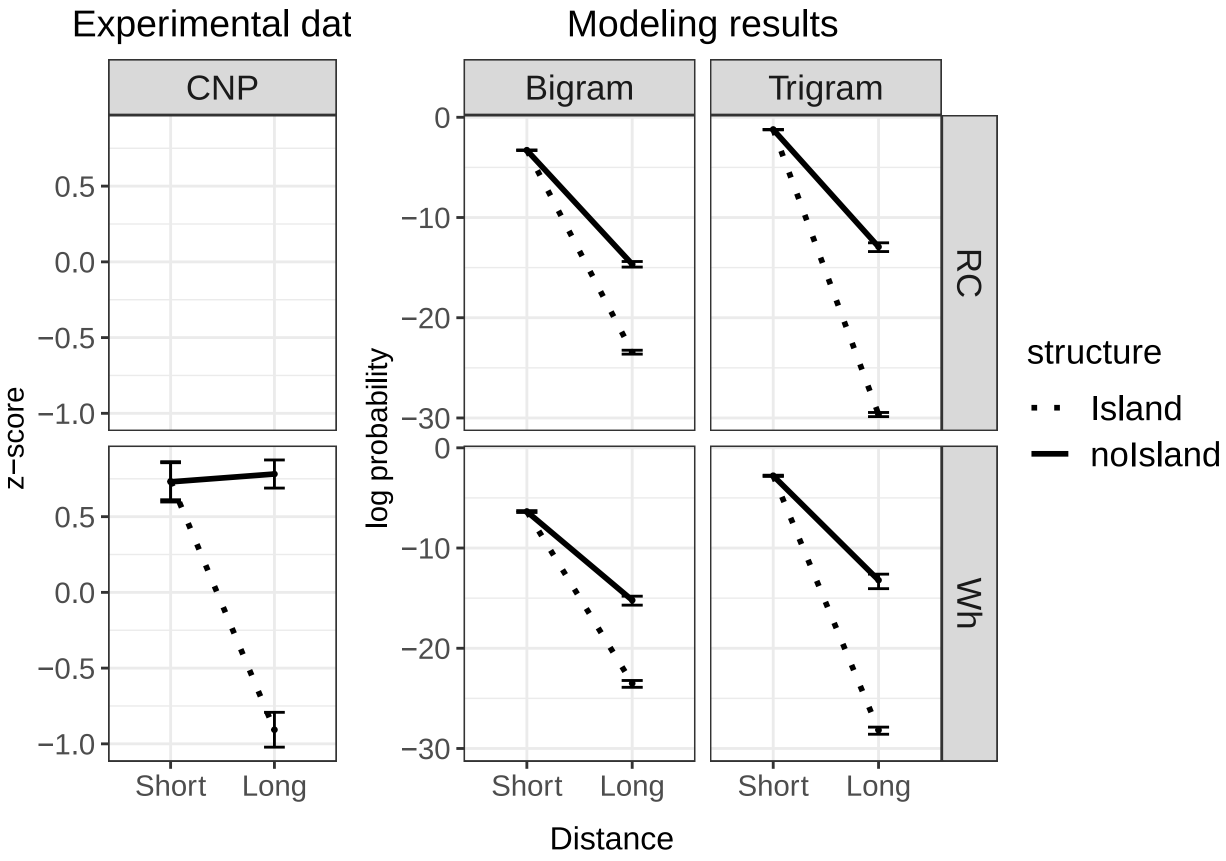
```
# Experimental data from other studies
other = read_excel("data/cond_means_other_comparisons.xlsx")

# Making sure that these variables are handled as factors
factors <- c("distance", "structure", "condition", "dependency")
other[factors] <- lapply(other[factors], factor)

# Z-score is a numeric variable
other$zscores = as.numeric(other$zscores)
```

```
exp_cnp = subset(other, condition=="CNP")
exp_whether_obj = subset(other, condition=="Whether-object")
exp_rc_pcomp = subset(other, condition=="RC-Pcomp")
```

```
m_cnp = plot_prob(model_cnp)
e_cnp = plot_zscore(exp_cnp)

plot_cnp = ggarrange(e_cnp + rremove("xlab"), m_cnp + rremove("xlab"), ncol=2, common.legend = TRUE,
                     legend = "right", widths = c(0.55, 1))
```

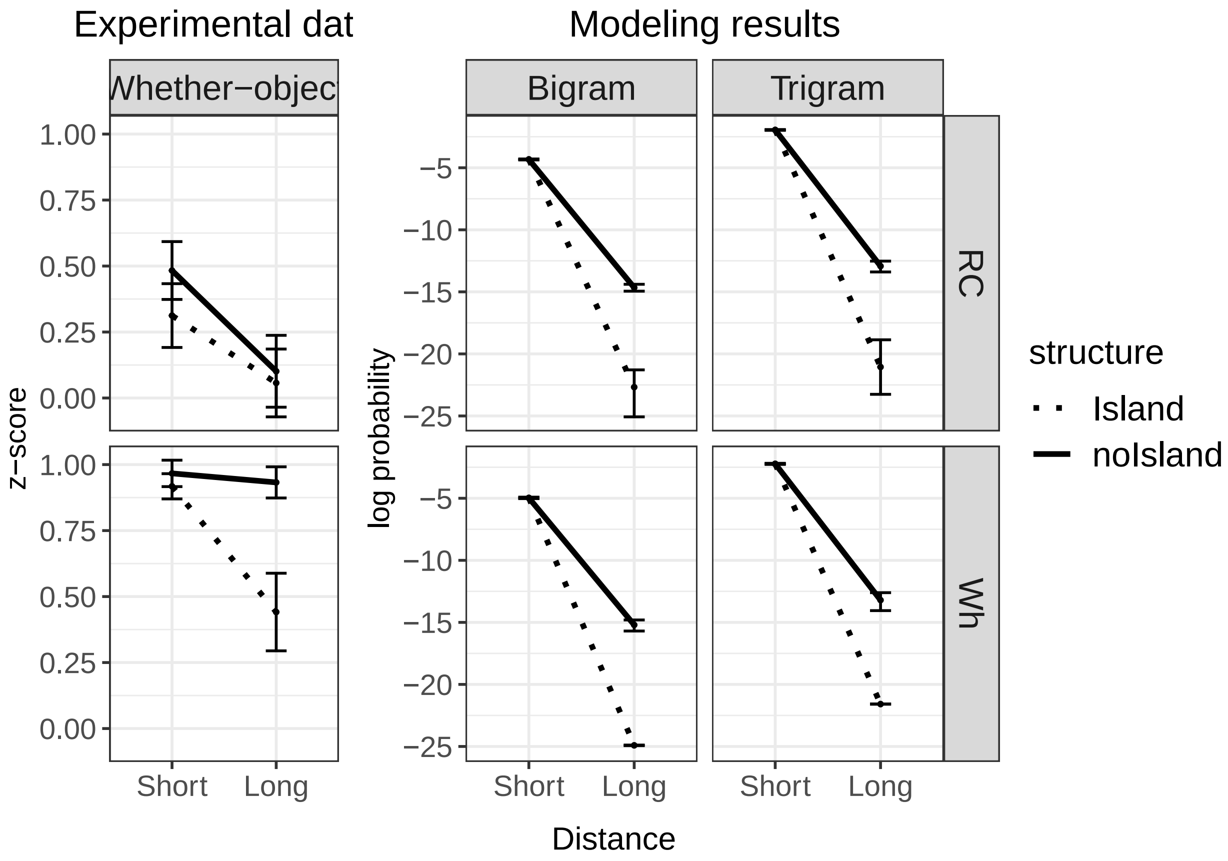
```
## Warning: Removed 4 rows containing missing values (`geom_point()`).

## Warning: Removed 4 rows containing missing values (`geom_line()`).

## Warning: Removed 4 rows containing missing values (`geom_point()`).
## Warning: Removed 4 rows containing missing values (`geom_line()`).
```

```
require(grid)   # for the textGrob() function
annotate_figure(plot_cnp, bottom = textGrob("Distance", gp = gpar(cex = 1)))
```



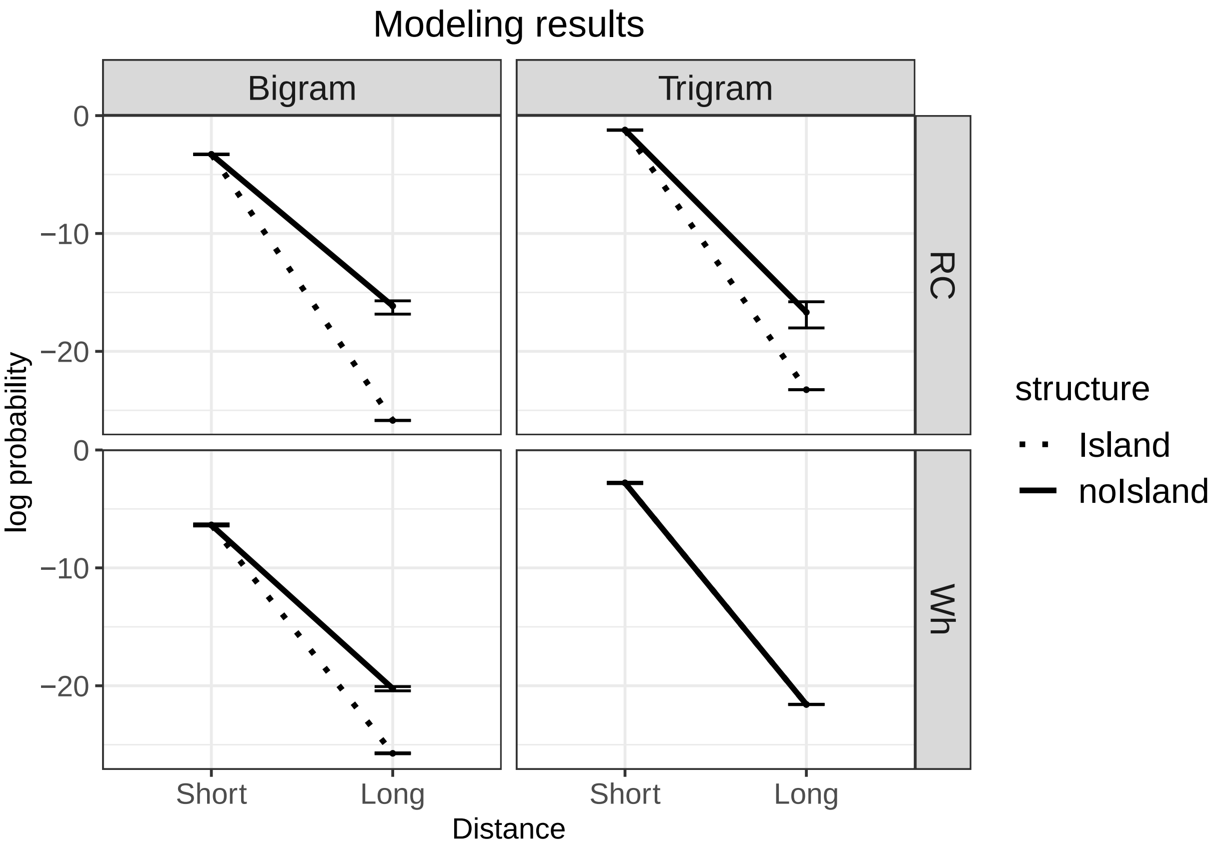
```
ggsave("plots/apdx_cnp.pdf", width = 10, height = 5)
```

```
m_whether_obj = plot_prob(model_whether_obj)
e_whether_obj = plot_zscore(exp_whether_obj)

plot_whether_obj = ggarrange(e_whether_obj + rremove("xlab"), m_whether_obj +
                               rremove("xlab"), ncol=2, common.legend = TRUE,
                    legend = "right", widths = c(0.55, 1))
require(grid)   # for the textGrob() function
annotate_figure(plot_whether_obj, bottom = textGrob("Distance", gp = gpar(cex = 1)))
```



```r
ggsave("plots/apdx_whether_obj.pdf", width = 10, height = 5)
```

```r
# No exp data for comparison
m_rc_subj = plot_prob(model_rc_subj)
m_rc_subj
```



```r
ggsave("plots/apdx_rc_subj.pdf", height = 5)
```

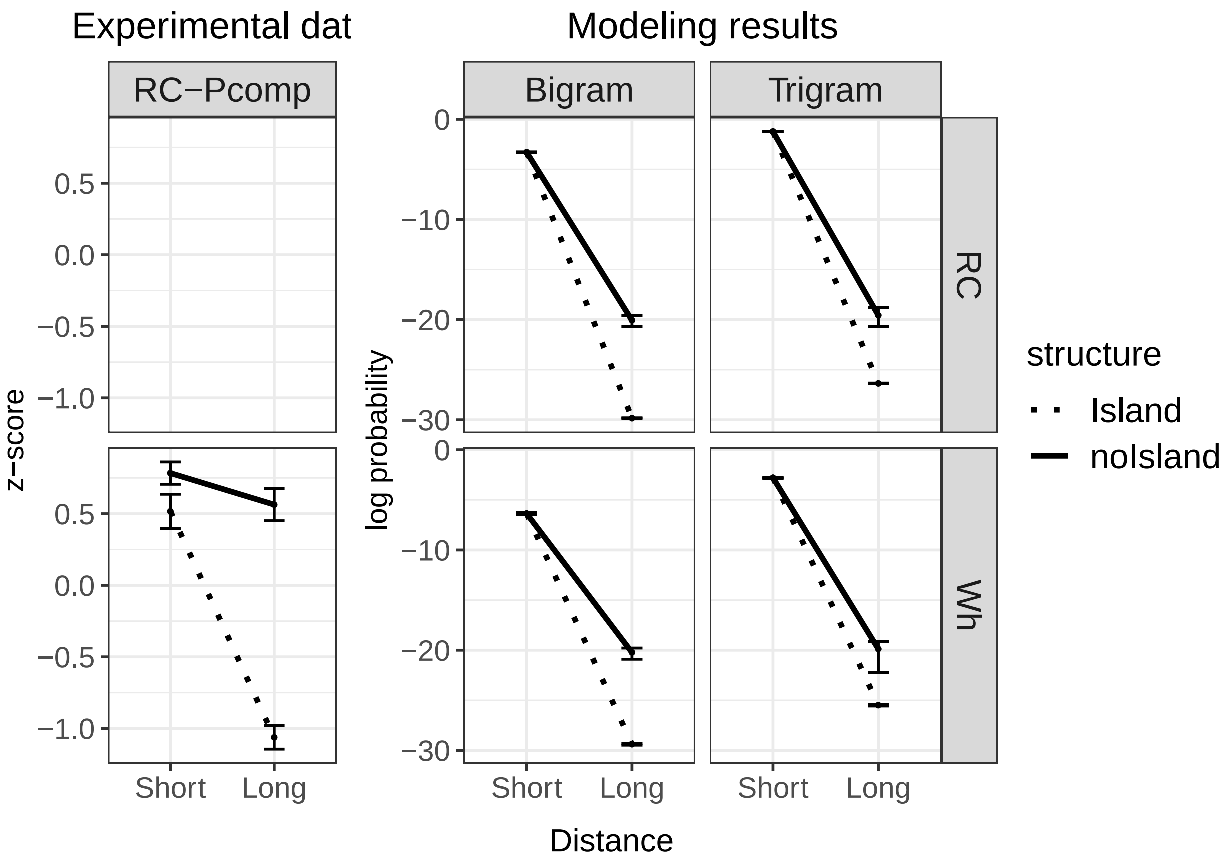
```
## Saving 6.5 x 5 in image
```

```r
e_rc_pcomp = plot_zscore(exp_rc_pcomp)
```

```r
m_rc_pcomp = plot_prob(model_rc_pcomp)
plot_rc_pcomp = ggarrange(e_rc_pcomp + rremove("xlab"), m_rc_pcomp + rremove("xlab"), ncol=2, common.le
                          legend = "right", widths = c(0.55, 1))
```

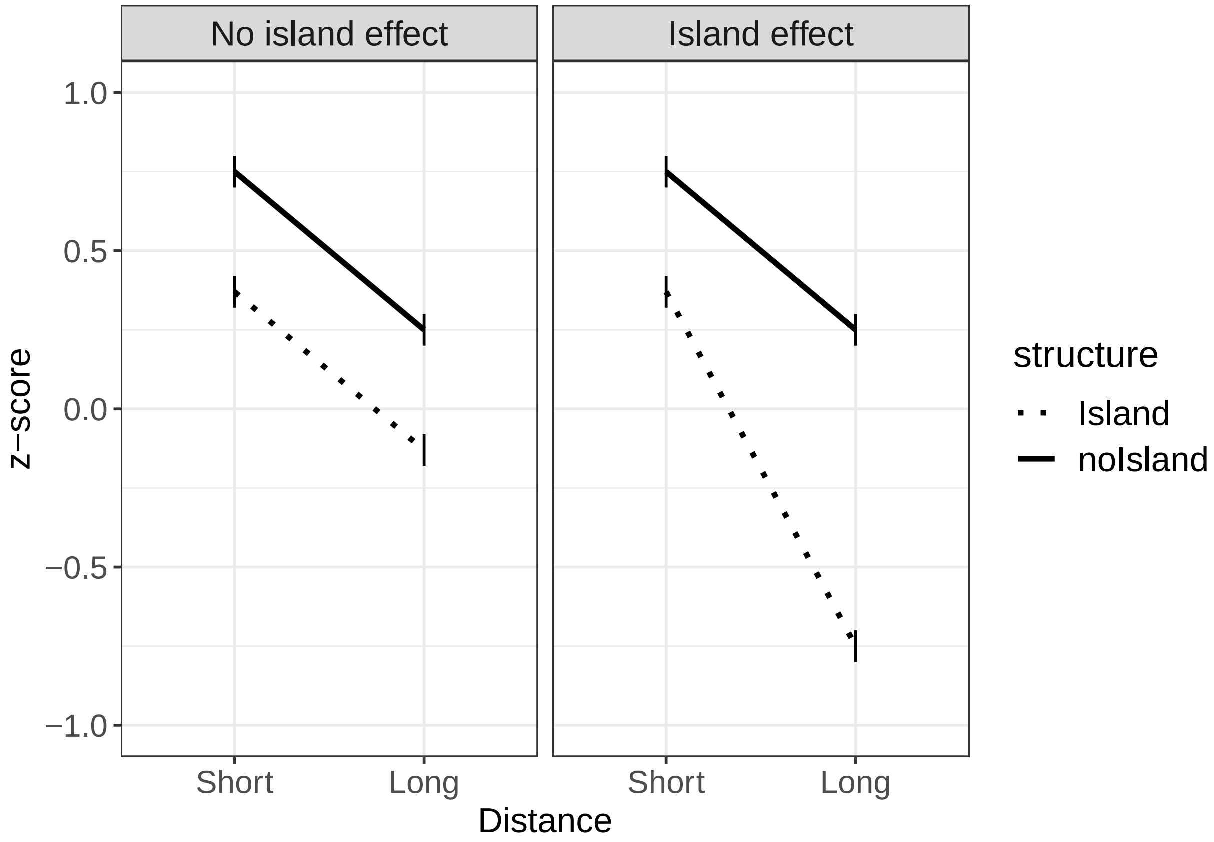
```
## Warning: Removed 4 rows containing missing values (`geom_point()`).

## Warning: Removed 4 rows containing missing values (`geom_line()`).

## Warning: Removed 4 rows containing missing values (`geom_point()`).

## Warning: Removed 4 rows containing missing values (`geom_line()`).
```

```r
require(grid)   # for the textGrob() function
annotate_figure(plot_rc_pcomp, bottom = textGrob("Distance", gp = gpar(cex = 1)))
```



```
ggsave("plots/apdx_rc_pcomp.pdf", width = 10, height = 5)
```

```
# Making the explanation graph
explanation_graph$zscores = as.numeric(explanation_graph$zscores)
explanation_graph$condition_f = factor(explanation_graph$condition,
                                        levels=c('No island effect','Island effect'))

ggplot(explanation_graph,
       aes(x=factor(ordered(distance, levels=c("Short","Long"))), y=zscores)) +
    geom_line(data=explanation_graph, aes(y=zscores, group=structure,
                              linetype=structure), linewidth=1) +
  geom_linerange(data=explanation_graph, aes(y=zscores, ymin=zscores-0.05,
                                         ymax=zscores+0.05), linewidth=0.5) +
  scale_linetype_manual(values=c("dotted", "solid")) +
    xlab("Distance") + ylab("z-score")+ facet_grid(~condition_f) +
  theme_bw() + theme(axis.text=element_text(size = 12),
                     axis.title=element_text(size = 13)) +
  ylim(-1,1) +
  theme(plot.title = element_text(hjust = 0.5, size = 15)) +  # title label
  theme(legend.text = element_text(size = 13), legend.title = element_text(size = 14)) +
  theme(strip.text = element_text(size = 13))  # facet label
```



```
ggsave("plots/exp_graph.pdf", width = 10, height = 4)
```

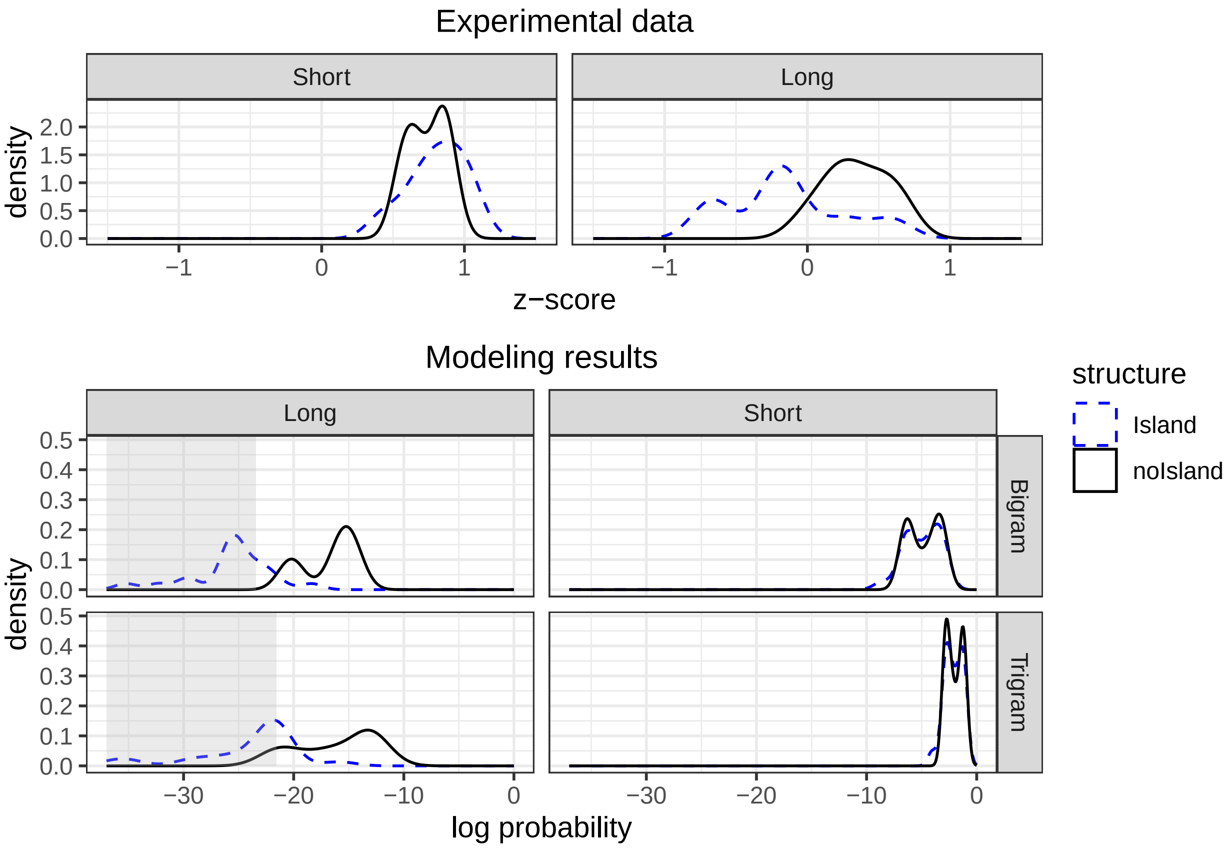
Additional explorations:

```
exp_data$distance = factor(exp_data$distance, levels = c("Short", "Long"), ordered = TRUE)
dens_exp = ggplot(data = exp_data, aes(x=zscores, group=structure))+
            geom_density(aes(color=structure, linetype=structure)) +
      scale_linetype_manual(values=c("dashed", "solid")) +
            scale_color_manual(values = c("blue","black"))+
      xlab('z-score') + facet_grid(~distance) +
      scale_x_continuous(limits = c(-1.5,1.5))+
      theme_bw() + ggtitle("Experimental data") +
      theme(plot.title = element_text(hjust = 0.5, size = 12))
```

```
xmins = c(-37, -37)
xmaxs = c(-23.43, -21.59)
rect_data <- data.frame(
  distance = as.factor(c("Long", "Long")),
  n_gram = as.factor(c("Bigram", "Trigram")),
  xmaxs = xmaxs,
  xmins = xmins)
dens_mod = ggplot(data = model_data, aes(x=log_probability, group=structure)) +
            geom_density(aes(color=structure, linetype=structure)) +
      scale_linetype_manual(values=c("dashed", "solid")) +
            scale_color_manual(values = c("blue","black")) +
      xlab('log probability') + facet_grid(n_gram~distance) +
      scale_x_continuous(limits = c(-37,0))+
      theme_bw() + ggtitle("Modeling results") +
      theme(plot.title = element_text(hjust = 0.5, size = 12)) +
    geom_rect(data = rect_data,
            aes(xmin = xmins, xmax = xmaxs, ymin = 0, ymax = Inf), alpha = 0.3, fill="grey", inherit.ae
```

```
dens_plot = ggarrange(dens_exp, dens_mod, ncol=1, common.legend = TRUE,
                      legend = "right", heights = c(0.7, 1.1))
```

```
dens_plot
```



```
ggsave("plots/density.pdf", width = 8, height = 6)
ggsave("plots/density.png", width = 8, height = 6)
```