REM

Interactions and Actors Data

Interactions between the team members for high-performance sessions:

- sender
- receiver
- time of the interaction
- type of interaction (dialogue act type)

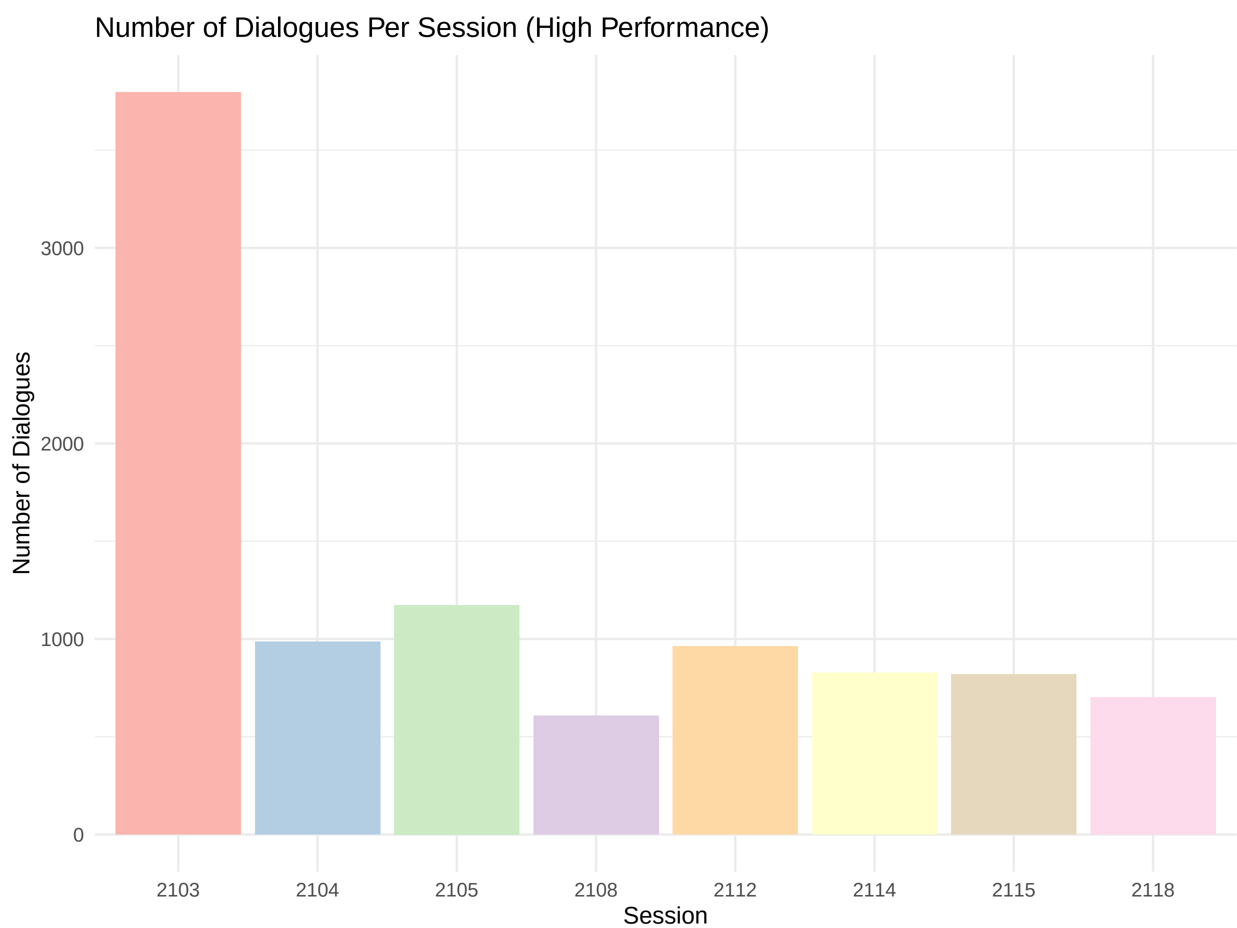
Actors Data:

- name
- gender

```r
# Interactions Data Frame (Edges)
high_perf_interactions <- readRDS("data/high_performance_sessions.RData") %>%
  select(session, sender_id, receiver_id, dialog, time)


interactions <- high_perf_interactions %>%
  mutate(
    sender_id = as.integer(sender_id),
    receiver_id = as.integer(receiver_id),
    dialog = as.factor(dialog)
  )



actors_attributes <- data.frame(
  id = 1:8,
  name = c("Igor", "Ashley", "Will", "Katya", "Saleh", "Oleg", "Vika", "Alex"),
  gender = c("male", "female", "male", "female", "male", "male", "female", "male")
)

# Create dummy variables for gender
dummyvars <- dummyVars(" ~ gender", data = actors_attributes)
actors_attributes <- cbind(actors_attributes, predict(dummyvars, actors_attributes)) %>%
  select(id, name, gendermale)
```

Summary by Session and Speaker

```r
session_dialogues <- high_perf_interactions %>%
  group_by(session) %>%
  summarise(n = n())


ggplot(session_dialogues, aes(x = factor(session), y = n, fill = factor(session))) +
  geom_bar(stat = "identity") +
  scale_fill_brewer(palette = "Pastel1") +
  labs(title = "Number of Dialogues Per Session (High Performance)",
       x = "Session",
       y = "Number of Dialogues",
       fill = "Session") +
  theme_minimal() +
  theme(legend.position = "none")
```

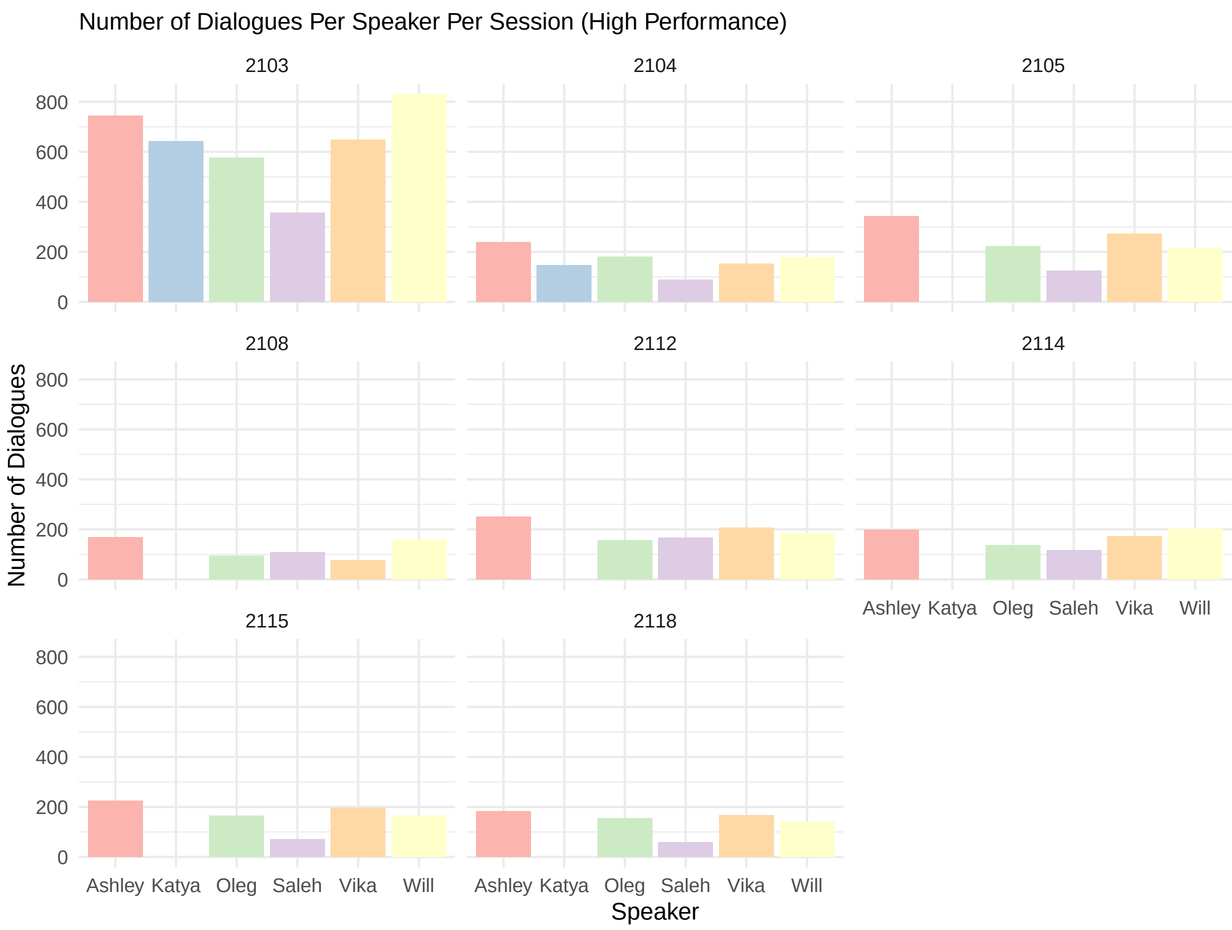



```r
dialogues_per_speaker_session <- high_perf_interactions %>%
  left_join(actors_attributes, by = c("sender_id" = "id")) %>%
  group_by(session, name) %>%
  summarise(number_of_dialogues = n(), .groups = 'drop') %>%
  arrange(session, desc(number_of_dialogues))

dialogues_summary_tibble <- as_tibble(dialogues_per_speaker_session)
```

```r
print(dialogues_summary_tibble)
```

```
## # A tibble: 42 x 3
##    session name   number_of_dialogues
##      <dbl> <chr>                <int>
##  1    2103 Will                   830
##  2    2103 Ashley                 744
##  3    2103 Vika                   649
##  4    2103 Katya                  642
##  5    2103 Oleg                   576
##  6    2103 Saleh                  356
##  7    2104 Ashley                 238
##  8    2104 Oleg                   181
##  9    2104 Will                   179
## 10    2104 Vika                   153
## # i 32 more rows
```

```r
ggplot(dialogues_summary_tibble, aes(x = name, y = number_of_dialogues, fill = name)) +
  geom_bar(stat = "identity") +
  facet_wrap(~session) +
  scale_fill_brewer(palette = "Pastel1") +
  labs(subtitle = "Number of Dialogues Per Speaker Per Session (High Performance)",
       x = "Speaker",
       y = "Number of Dialogues",
       fill = "Speaker") +
  theme_minimal() +
  theme(legend.position = "none")
```



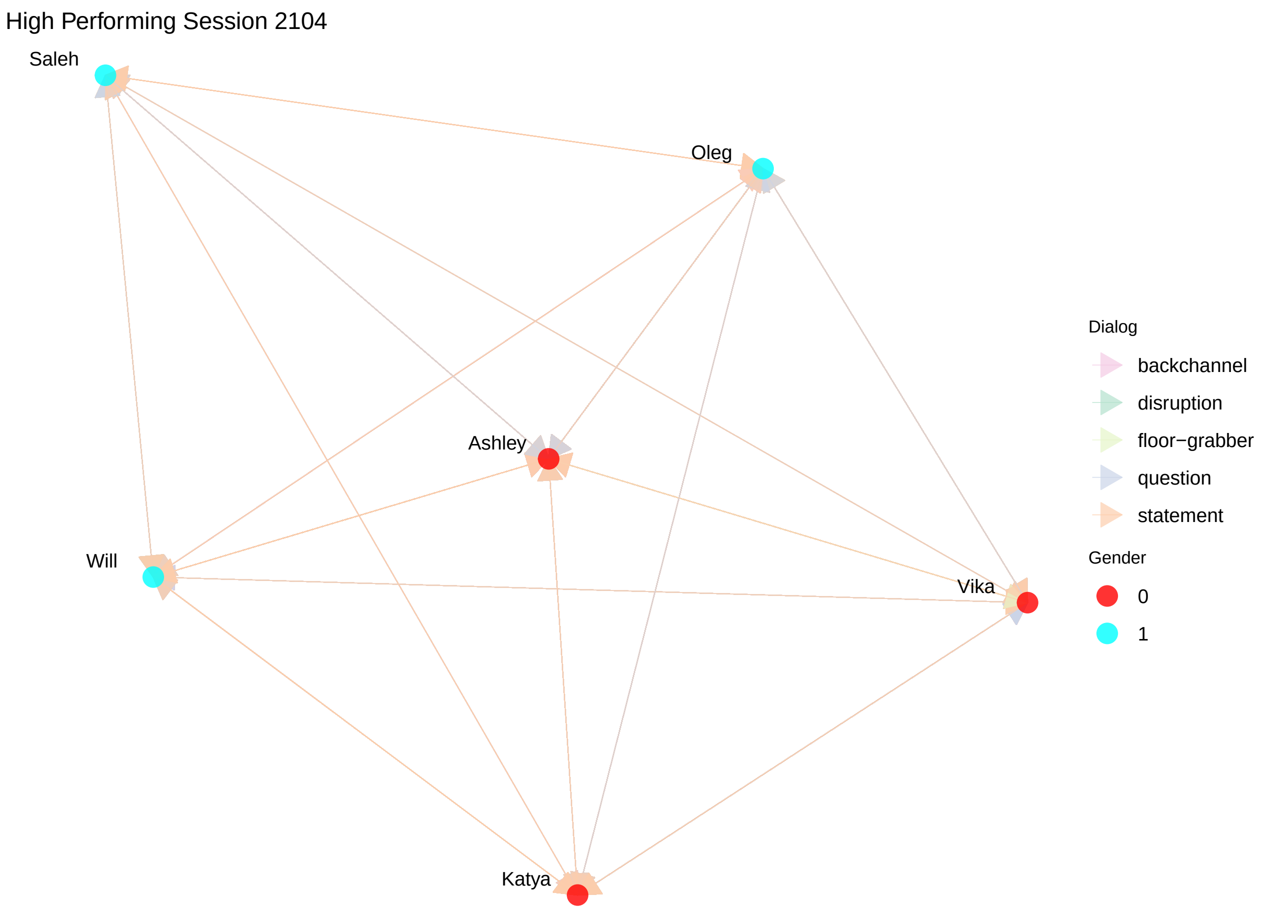

```
target_session <- 2104

high_perf_interactions %>% filter(receiver_id != 0) %>% filter(session == target_session) %>% select(-se

actors_attributes %>% filter(id %in% interactions$sender_id) %>% filter(id %in% interactions$receiver_i
head(actors_attributes)
```

```
##   id   name gendermale
## 2  2 Ashley          0
## 3  3   Will          1
## 4  4  Katya          0
## 5  5  Saleh          1
## 6  6   Oleg          1
## 7  7   Vika          0
```

```
g_subset <- graph_from_data_frame(interactions, directed = TRUE, vertices = data.frame(actors_attribute


V(g_subset)$gender <- actors_attributes$gender[match(V(g_subset)$name, actors_attributes$name)]
V(g_subset)$name <- actors_attributes$name[match(V(g_subset)$name, actors_attributes$name)]
```

```r
dialog_colors <- RColorBrewer::brewer.pal(n = length(unique(interactions$dialog)), name = "Pastel2")
dialog_color_map <- setNames(dialog_colors, unique(interactions$dialog))


ggraph(g_subset, layout = 'fr') +
  geom_edge_link(aes(color = dialog), alpha = 0.7, edge_width = .2, lineend = "butt", arrow = arrow(type
  scale_edge_color_manual(values = dialog_color_map) +
  geom_node_point(aes(color = factor(gender)), size = 4, alpha = 0.8) +
  geom_node_text(aes(label = name), repel = TRUE,  color = "black", size = 3, vjust = 1, nudge_x = -.02)
  scale_color_manual(values = c('0' = 'red', '1' = 'cyan')) +
  theme_void() +
  labs(subtitle = "High Performing Session 2104", color = "Gender", edge_color = "Dialog") +
  theme(legend.position = "right", legend.title = element_text(size = 8))
```

Dialogue Flow Illustration




```r
degree_stats <- degree(g_subset, mode = "all")
```

```r
betweenness_stats <- betweenness(g_subset, directed = TRUE)

network_stats_summary <- data.frame(
  name = V(g_subset)$name,
  degree = degree_stats
)


print(network_stats_summary)
```

Degree Distribution

```
##          name degree
## Ashley Ashley    473
## Will     Will    358
## Katya   Katya    294
## Saleh   Saleh    178
## Oleg     Oleg    363
## Vika     Vika    306
```

```r
influencers <- network_stats_summary %>%
  arrange(desc(degree)) %>%
  head(3)
print(influencers)
```

```
##          name degree
## Ashley Ashley    473
## Oleg     Oleg    363
## Will     Will    358
```

- **Degree**: This is the number of direct connections a node has.
  - Ashley has the highest degree (473), or the greatest direct connections in the network.
- **Closeness**: This measures how quickly a node can access all other nodes in the network. Higher values represent shorter paths to all other nodes.
  - Ashley, with a closeness of 1.0000000, is the quickest to reach all other nodes.

Check for Isolates, Connectivity, and Directionality

```r
isolates <- which(degree(g_subset) == 0)
if (length(isolates) > 0) {
  print(V(g_subset)$name[isolates])
}


is.fully.connected <- is_connected(g_subset)
print(is.fully.connected)
```

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

```r
is.directed <- is_directed(g_subset)
print(is.directed)
```

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

```r
in_degree_stats <- degree(g_subset, mode = "in")
out_degree_stats <- degree(g_subset, mode = "out")
total_degree_stats <- in_degree_stats + out_degree_stats
rbind(in_degree_stats, out_degree_stats, total_degree_stats)
```

```
##                    Ashley Will Katya Saleh Oleg Vika
## in_degree_stats       236  179   147    89  182  153
## out_degree_stats      237  179   147    89  181  153
## total_degree_stats    473  358   294   178  363  306
```

REM Analysis

```r
interactions$time<-as.numeric(interactions$time)

# Create the REM data set
REM.data <- createRemDataset(
  data = interactions,
  sender = interactions$sender_id,
  target = interactions$receiver_id,
  eventSequence = interactions$time,
  eventAttribute = interactions$dialog,
  atEventTimesOnly = TRUE,
  untilEventOccurrs = TRUE,
  includeAllPossibleEvents = FALSE,
  returnInputData = FALSE
)

#save as RDS
#saveRDS(REM.data, "data/REM_data_onlyevent.RDS")
```

```r
readRDS("data/REM_data.RDS") -> REM.data
```

```r
# Check the structure of the REM.data
str(REM.data)
```

```
## 'data.frame':    90290 obs. of  12 variables:
##  $ target          : chr  "2" "2" "2" "2" ...
##  $ sender          : chr  "2" "3" "3" "6" ...
##  $ eventID         : chr  "eventID1" "eventID96" "eventID96" "eventID969" ...
##  $ eventTime       : num  1 38 39 959 960 961 962 179 180 181 ...
##  $ eventDummy      : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ eventAtRiskFrom : num  1 1 1 949 949 949 949 1 1 1 ...
##  $ eventAtRiskUntil: num  1 96 96 969 969 969 969 199 199 199 ...
```

```
##  $ eventAttribute  : chr  "disruption" "statement" "statement" "statement" ...
##  $ name.x          : chr  "Ashley" "Will" "Will" "Oleg" ...
##  $ gendermale.x    : num  0 1 1 1 1 1 1 0 0 0 ...
##  $ name.y          : chr  "Ashley" "Ashley" "Ashley" "Ashley" ...
##  $ gendermale.y    : num  0 0 0 0 0 0 0 0 0 0 ...
```

```r
head(REM.data)
```

```
##   target sender    eventID eventTime eventDummy eventAtRiskFrom
## 1      2      2   eventID1         1          1               1
## 2      2      3  eventID96        38          0               1
## 3      2      3  eventID96        39          0               1
## 4      2      6 eventID969       959          0             949
## 5      2      6 eventID969       960          0             949
## 6      2      6 eventID969       961          0             949
##   eventAtRiskUntil eventAttribute name.x gendermale.x name.y gendermale.y
## 1                1     disruption Ashley            0 Ashley            0
## 2               96      statement   Will            1 Ashley            0
## 3               96      statement   Will            1 Ashley            0
## 4              969      statement   Oleg            1 Ashley            0
## 5              969      statement   Oleg            1 Ashley            0
## 6              969      statement   Oleg            1 Ashley            0
```

```r
surv_object <- Surv(time = REM.data$eventTime, event = REM.data$eventDummy)


cox_model <- coxph(surv_object ~ sender + target, data = REM.data)


summary(cox_model)
```

```
## Call:
## coxph(formula = surv_object ~ sender + target, data = REM.data)
##
##   n= 90290, number of events= 986
##
##            coef exp(coef) se(coef)      z Pr(>|z|)
## sender3 -0.4268    0.6526   0.1033 -4.131 3.62e-05 ***
## sender4 -0.2163    0.8055   0.1072 -2.018   0.0436 *
## sender5 -0.8191    0.4408   0.1267 -6.463 1.03e-10 ***
## sender6 -0.4809    0.6182   0.1024 -4.697 2.63e-06 ***
## sender7 -0.4295    0.6509   0.1070 -4.015 5.94e-05 ***
## target3 -0.4370    0.6460   0.1033 -4.230 2.34e-05 ***
## target4 -0.1341    0.8745   0.1077 -1.245   0.2131
## target5 -0.7362    0.4789   0.1270 -5.798 6.72e-09 ***
## target6 -0.5767    0.5617   0.1027 -5.618 1.94e-08 ***
## target7 -0.1409    0.8686   0.1053 -1.339   0.1807
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##         exp(coef) exp(-coef) lower .95 upper .95
## sender3    0.6526      1.532    0.5329    0.7991
## sender4    0.8055      1.241    0.6529    0.9938
```

```
## sender5    0.4408      2.268    0.3439    0.5651
## sender6    0.6182      1.618    0.5058    0.7556
## sender7    0.6509      1.536    0.5278    0.8026
## target3    0.6460      1.548    0.5275    0.7910
## target4    0.8745      1.143    0.7081    1.0800
## target5    0.4789      2.088    0.3734    0.6143
## target6    0.5617      1.780    0.4593    0.6869
## target7    0.8686      1.151    0.7067    1.0676
##
## Concordance= 0.606  (se = 0.011 )
## Likelihood ratio test= 93.47  on 10 df,   p=1e-15
## Wald test            = 90.93  on 10 df,   p=4e-15
## Score (logrank) test = 92.43  on 10 df,   p=2e-15
```

```
model2_event <- coxph(surv_object ~ eventAttribute, data = REM.data)
summary(model2_event)
```

```
## Call:
## coxph(formula = surv_object ~ eventAttribute, data = REM.data)
##
##   n= 90290, number of events= 986
##
##                                coef exp(coef) se(coef)     z Pr(>|z|)
## eventAttributedisruption    0.1617    1.1755   0.2717 0.595    0.552
## eventAttributefloor-grabber 0.3357    1.3989   0.2273 1.477    0.140
## eventAttributequestion      0.9340    2.5447   0.1853 5.041 4.63e-07 ***
## eventAttributestatement     1.4839    4.4099   0.1789 8.294  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                             exp(coef) exp(-coef) lower .95 upper .95
## eventAttributedisruption        1.176     0.8507    0.6901     2.002
## eventAttributefloor-grabber     1.399     0.7148    0.8961     2.184
## eventAttributequestion          2.545     0.3930    1.7698     3.659
## eventAttributestatement         4.410     0.2268    3.1056     6.262
##
## Concordance= 0.641  (se = 0.009 )
## Likelihood ratio test= 207  on 4 df,   p=<2e-16
## Wald test            = 172.3  on 4 df,   p=<2e-16
## Score (logrank) test = 190.6  on 4 df,   p=<2e-16
```

```
model3_snd_event <- coxph(surv_object ~ sender + eventAttribute, data = REM.data)
summary(model3_snd_event)
```

```
## Call:
## coxph(formula = surv_object ~ sender + eventAttribute, data = REM.data)
##
##   n= 90290, number of events= 986
##
##                                 coef exp(coef) se(coef)      z Pr(>|z|)
## sender3                     -0.32011   0.72607  0.09953 -3.216 0.001299 **
## sender4                     -0.36260   0.69586  0.10563 -3.433 0.000598 ***
## sender5                     -0.76721   0.46431  0.12467 -6.154 7.55e-10 ***
```

```
## sender6                      -0.49032   0.61243  0.09967 -4.919 8.68e-07 ***
## sender7                      -0.32726   0.72089  0.10440 -3.135 0.001720 **
## eventAttributedisruption      0.23939   1.27048  0.27340  0.876 0.381237
## eventAttributefloor-grabber  0.38767   1.47355  0.22795  1.701 0.088995 .
## eventAttributequestion        1.04592   2.84602  0.18777  5.570 2.55e-08 ***
## eventAttributestatement       1.58097   4.85967  0.18048  8.760  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                              exp(coef) exp(-coef) lower .95 upper .95
## sender3                         0.7261     1.3773    0.5974    0.8825
## sender4                         0.6959     1.4371    0.5657    0.8559
## sender5                         0.4643     2.1538    0.3637    0.5928
## sender6                         0.6124     1.6328    0.5038    0.7446
## sender7                         0.7209     1.3872    0.5875    0.8846
## eventAttributedisruption        1.2705     0.7871    0.7434    2.1711
## eventAttributefloor-grabber     1.4735     0.6786    0.9426    2.3035
## eventAttributequestion          2.8460     0.3514    1.9697    4.1122
## eventAttributestatement         4.8597     0.2058    3.4118    6.9220
##
## Concordance= 0.665  (se = 0.01 )
## Likelihood ratio test= 254.2  on 9 df,   p=<2e-16
## Wald test            = 219.7  on 9 df,   p=<2e-16
## Score (logrank) test = 238.7  on 9 df,   p=<2e-16
```

- The `Concordance` statistic is a measure of the model's predictive ability, with 0.5 indicating no predictive ability and 1 indicating perfect prediction.

Survival Analysis

- How different factors (like senders, targets, event attributes) affect the likelihood of events:
- `coef` in the model indicate the *logarithmic* change in hazard rates for different categories compared to a *baseline*. A positive coefficient indicates an increased hazard (or risk) of the event occurring when the variable increases, while a negative coefficient indicates a decreased hazard.
- `exp(coef)` represents the *hazard ratio*, which explains the effect size. values greater than 1 indicate an increased hazard, values less than 1 indicate a decreased hazard.
- The p-values (`Pr(>|z|)`) help determine the significance of the predictors.

Result Analysis:

Model 1: `sender + target` as predictors

predictors are the sender and target IDs of the events.

- **Senders 3, 4, 5, 6, and 7** have significant negative coefficients, suggesting that events sent by these individuals are less likely to occur compared to the baseline.
- **Target 3 and 6** have significant negative coefficients, indicating events targeting these individuals are less likely to occur than the baseline target.
- The p-values for the above mentioned actors are all below 0.05.
- The overall model shows good predictive power with a concordance index of 0.606.

Model 2: `eventAttribute` as predictors

- **`question` and `statement` types of dialog** have a positive and significant effect on the hazard, indicating these types of events are more likely to occur. Their coefficients are positive with low p-values (< 2e-16 for `statement`, 4.63e-07 for `question`), showing strong evidence for their influence.
- The `disruption` and `floor-grabber` types do not show a significant effect since their p-values are above 0.05.
- The model has a concordance index of 0.641, indicating a reasonably good fit.

Model 3: `sender` + `eventAttribute` - senders and dialog types as predictors.

- Similar to Model 1, **senders 3, 4, 5, 6, and 7** are significant predictors with negative coefficients, meaning events from these senders are less likely to happen compared to the reference sender.
- As in Model 2, **`question` and `statement`** types of dialog are significant with positive coefficients, which means these event types are more likely to occur.
- The coefficients for `disruption` and `floor-grabber` are not significant in this model as well
- The concordance index is 0.665, the **highest** among the three models, suggesting this model has the best predictive ability.

Summary

- **Sender Effect**: Negative coefficients for all senders indicate they all decrease the event's hazard compared to a baseline, with `sender5` showing a significant reduction.
- **Target Effect**: Similarly, most targets also reduce hazard rates, except `target4` and `target7` where the effect is not significant.
- **Event Attributes** significantly increases the hazard of an event occurring.
  - The `eventAttribute` variables have significant coefficients for `question` and `statement`, indicating that these attributes are associated with the occurrence of events.
  - The hazard ratio of `statement` is about 4.41, suggesting its strong association with the occurrence of events...Is the effect of "statement" on event occurrences is genuinely significant, or is it inflated by the volume?
  - `question` turns out to be important!