EDA for Final Project

Yinyu Yao

2025-11-13

```r
library(tidytuesdayR)
```

```
## Warning: package 'tidytuesdayR' was built under R version 4.4.3
```

```r
tuesdata = tidytuesdayR::tt_load(2025, week = 13)
```

```
## ---- Compiling #TidyTuesday Information for 2025-04-01 ----
## --- There is 1 file available ---
##
##
## -- Downloading files -----------------------------------------------------------
##
##   1 of 1: "pokemon_df.csv"
```

```r
pokemon_df = tuesdata$pokemon_df
pokemon_df = subset(pokemon_df, select= -c(id, species_id, color_1, color_2, color_f, egg_group_1, egg_g
```

```r
# Count rows with at least one missing value
pokemon_df_without_type2_and_generation_id =  subset(pokemon_df, select= -c(type_2, generation_id))
row_with_na = !complete.cases(pokemon_df_without_type2_and_generation_id)
n_sparse <- sum(row_with_na)
pct_sparse <- n_sparse / nrow(pokemon_df_without_type2_and_generation_id)

cat("Number of rows with at least one N.A.(not count type2 or generation_id):", n_sparse, "\n")
```

```
## Number of rows with at least one N.A.(not count type2 or generation_id): 0
```

```r
empty_type2 <- sum(is.na(pokemon_df$type_2))
empty_generation_id <- sum(is.na(pokemon_df$generation_id))

unique_generation_id <- sort(unique(pokemon_df$generation_id))
n_unique_gen <- length(unique_generation_id)

n_gen1 <- sum(pokemon_df$generation_id == 1, na.rm = TRUE)

cat("Number of rows with missing type_2:", empty_type2, "\n")
```

```
## Number of rows with missing type_2: 439
```

```r
cat("Number of rows with missing generation_id:", empty_generation_id, "\n")
```

```
## Number of rows with missing generation_id: 147
```

```r
cat("Unique generation IDs:", paste(unique_generation_id, collapse = ", "), "\n")
```

```
## Unique generation IDs: 1, 2, 3, 4, 5, 6, 7
```

```r
gen_tab <- as.data.frame(table(pokemon_df$generation_id, useNA = "ifany"))
names(gen_tab) <- c("generation_id", "count")
knitr::kable(gen_tab, caption = "Counts of Pokémon by generation_id")
```

Table 1: Counts of Pokémon by generation_id

generation_id	count
1	151
2	100
3	135
4	107
5	156
6	72
7	81
NA	147

```r
type_tab <- as.data.frame(table(pokemon_df$type_1, useNA = "ifany"))
names(type_tab) <- c("type_1", "count")
knitr::kable(type_tab, caption = "Counts of Pokémon by type_1")
```

Table 2: Counts of Pokémon by type_1

type_1	count
bug	79
dark	37
dragon	39
electric	61
fairy	19
fighting	31
fire	59
flying	4
ghost	40
grass	84
ground	36
ice	29
normal	111
poison	35
psychic	64
rock	65
steel	30

type_1	count
water	126

```r
# conditional distribution of type2 given type1 (only for rows with type2)
pokemon_types <- pokemon_df |>
  filter(!is.na(type_2))

type_pair_counts <- pokemon_types |>
  count(type_1, type_2, name = "n") |>
  group_by(type_1) |>
  mutate(
    row_total = sum(n),
    prop = n / row_total
  ) |>
  arrange(type_1, desc(prop)) |>
  ungroup()
```

```r
readr::write_csv(type_pair_counts,
                 "tables/type1_type2_joint_counts_proportions.csv")

# Print first few rows (most informative combos)
knitr::kable(
  head(type_pair_counts, 30),
  digits = 3,
  caption = "Joint counts and row-wise proportions P(type_2 | type_1) (top 30 rows)."
)
```

Table 3: Joint counts and row-wise proportions P(type_2 | type_1) (top 30 rows).

type_1	type_2	n	row_total	prop
bug	flying	14	61	0.230
bug	poison	12	61	0.197
bug	steel	7	61	0.115
bug	grass	6	61	0.098
bug	electric	5	61	0.082
bug	fighting	4	61	0.066
bug	rock	3	61	0.049
bug	water	3	61	0.049
bug	fairy	2	61	0.033
bug	fire	2	61	0.033
bug	ground	2	61	0.033
bug	ghost	1	61	0.016
dark	flying	5	25	0.200
dark	dragon	4	25	0.160
dark	fire	3	25	0.120
dark	normal	3	25	0.120
dark	fighting	2	25	0.080
dark	ghost	2	25	0.080
dark	ice	2	25	0.080
dark	psychic	2	25	0.080

type_1	type_2	n	row_total	prop
dark	steel	2	25	0.080
dragon	ground	8	27	0.296
dragon	flying	6	27	0.222
dragon	psychic	4	27	0.148
dragon	fighting	3	27	0.111
dragon	ice	3	27	0.111
dragon	electric	1	27	0.037
dragon	fairy	1	27	0.037
dragon	fire	1	27	0.037
electric	flying	6	21	0.286

```r
# For each type_1,  the most common type_2 (ties allowed)
type2_pref <- type_pair_counts |>
  group_by(type_1) |>
  slice_max(prop, n = 1, with_ties = TRUE) |>
  arrange(type_1, desc(prop)) |>
  ungroup()

readr::write_csv(type2_pref,
                 "tables/type2_preference_by_type1.csv")

knitr::kable(
  type2_pref,
  digits = 3,
  caption = "Most frequent secondary type(s) for each primary type (type_1)."
)
```

Table 4: Most frequent secondary type(s) for each primary type (type_1).

type_1	type_2	n	row_total	prop
bug	flying	14	61	0.230
dark	flying	5	25	0.200
dragon	ground	8	27	0.296
electric	flying	6	21	0.286
fairy	flying	2	2	1.000
fighting	psychic	3	9	0.333
fire	fighting	7	28	0.250
fire	flying	7	28	0.250
flying	dragon	2	2	1.000
ghost	grass	11	30	0.367
grass	poison	15	45	0.333
ground	flying	4	21	0.190
ice	ground	3	14	0.214
ice	water	3	14	0.214
normal	flying	27	44	0.614
poison	dark	5	20	0.250
psychic	fairy	7	23	0.304
psychic	flying	7	23	0.304
rock	flying	18	53	0.340

type_1	type_2	n	row_total	prop
steel	psychic	7	25	0.280
water	ground	10	60	0.167

```r
# Chi-squared test: are type_1 and type_2 independent?
tab_t1_t2 <- table(
  type_1 = pokemon_types$type_1,
  type_2 = pokemon_types$type_2
)

chi_res <- chisq.test(tab_t1_t2)
```

```
## Warning in stats::chisq.test(x, y, ...): Chi-squared approximation may be
## incorrect
```

```r
chi_res  # printed summary
```

```
##
##  Pearson's Chi-squared test
##
## data:  tab_t1_t2
## X-squared = 698.64, df = 289, p-value < 2.2e-16
```

```r
# Tidy-ish version for saving
chi_tbl <- tibble(
  statistic = chi_res$statistic,
  df        = chi_res$parameter,
  p_value   = chi_res$p.value
)

readr::write_csv(chi_tbl,
                 "tables/chisq_type1_type2_independence.csv")

knitr::kable(
  chi_tbl,
  caption = "Chi-squared test of independence between type_1 and type_2."
)
```

Table 5: Chi-squared test of independence between type_1 and type_2.

statistic	df	p_value
698.6441	289	0

```r
# Distribution of Pokemon abilities by generation_id
generation_id_f = fct_explicit_na(as.factor(pokemon_df$generation_id), na_level = "Unknown")
```

```
## Warning: `fct_explicit_na()` was deprecated in forcats 1.0.0.
## i Please use `fct_na_value_to_level()` instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

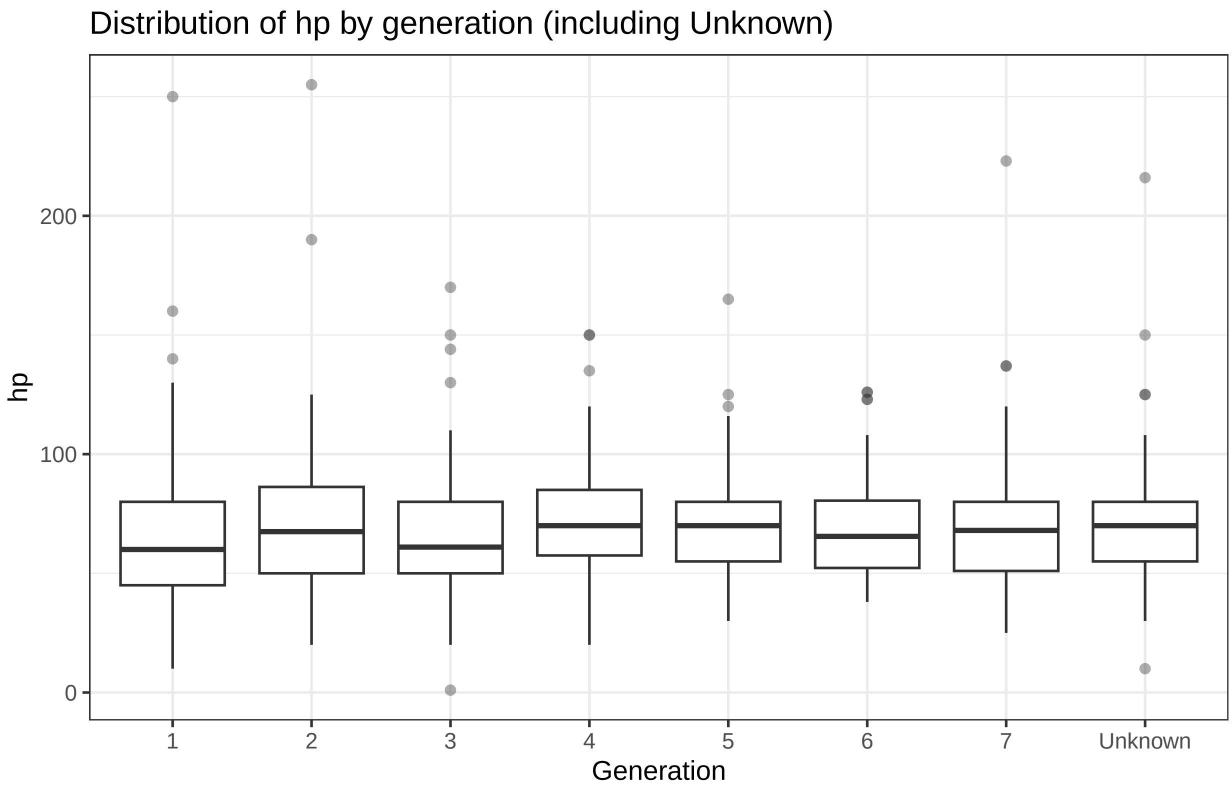
```
pokemon_df <- pokemon_df |>
  janitor::clean_names() |>
  mutate(
    base_stat_total = hp + attack + defense +
      special_attack + special_defense + speed,generation_id_f)

stat_vars <- c("hp", "attack", "defense",
               "special_attack", "special_defense", "speed")


# ---- Summary table by generation (including Unknown) ----
gen_stats_all <- pokemon_df |>
  group_by(generation_id_f) |>
  summarise(
    n = n(),
    across(
      all_of(stat_vars),
      list(mean = ~mean(., na.rm = TRUE),
           sd   = ~sd(.,   na.rm = TRUE)),
      .names = "{.col}_{.fn}"
    )
  ) |>
  arrange(generation_id_f)

readr::write_csv(gen_stats_all,
                 "tables/gen_stats_by_generation_all_stats.csv")

knitr::kable(
  gen_stats_all,
  digits = 1,
  caption = "Means and standard deviations of stats by generation (including Unknown)."
)
```

Table 6: Means and standard deviations of stats by generation (including Unknown).

generation_id_f	n	hp_mean	hp_sd	attack_mean	attack_sd	defense_mean	defense_sd	special_attack_mean	special_attack_sd	special_defense_mean	special_defense_sd	speed_mean	speed_sd
1	151	64.2	28.6	72.9	26.8	68.2	26.9	67.1	28.5	66.1	24.2	69.1	27.0
2	100	71.0	31.2	68.3	28.4	69.7	35.2	64.5	25.6	72.3	31.5	61.4	27.2
3	135	65.7	25.2	73.1	30.4	69.0	31.1	67.9	28.3	66.5	28.5	61.6	26.9
4	107	73.1	24.7	80.2	30.9	75.2	30.7	73.3	31.2	74.5	27.8	69.5	27.6
5	156	70.3	21.6	81.0	29.4	71.2	23.0	69.2	29.8	67.3	21.9	66.6	28.2
6	72	68.9	21.7	72.5	25.6	75.2	31.7	72.5	28.0	74.7	30.8	65.7	25.9
7	81	70.7	28.2	83.2	32.6	77.0	29.9	73.5	33.4	74.8	29.3	64.5	29.1
Unknown	147	70.1	25.1	98.9	37.5	87.7	34.4	92.0	42.0	84.6	26.2	87.3	31.4

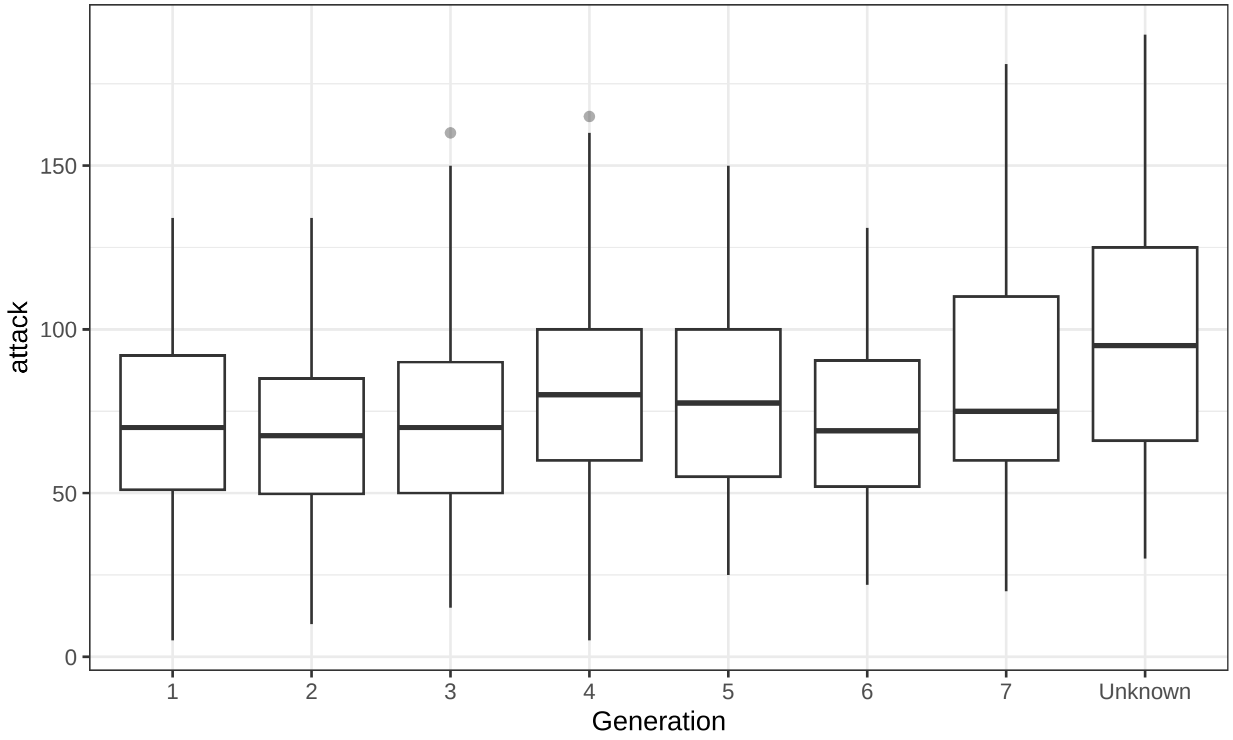
```r
# Generate and save boxplots for each stat by generation
for (s in stat_vars) {
  p <- ggplot(pokemon_df,
              aes(x = generation_id_f, y = .data[[s]])) +
    geom_boxplot(outlier.alpha = 0.4) +
    labs(
      title = paste0("Distribution of ", s, " by generation (including Unknown)"),
      x = "Generation",
      y = s
    ) +
    theme_bw()

  file_name <- paste0("figures/box_", s, "_by_generation.png")
  ggsave(file_name, p, width = 7, height = 4.5, dpi = 150)

  cat("Saved:", file_name, "\n")
  print(p)  # show in the knitted report
}
```

```
## Saved: figures/box_hp_by_generation.png
```



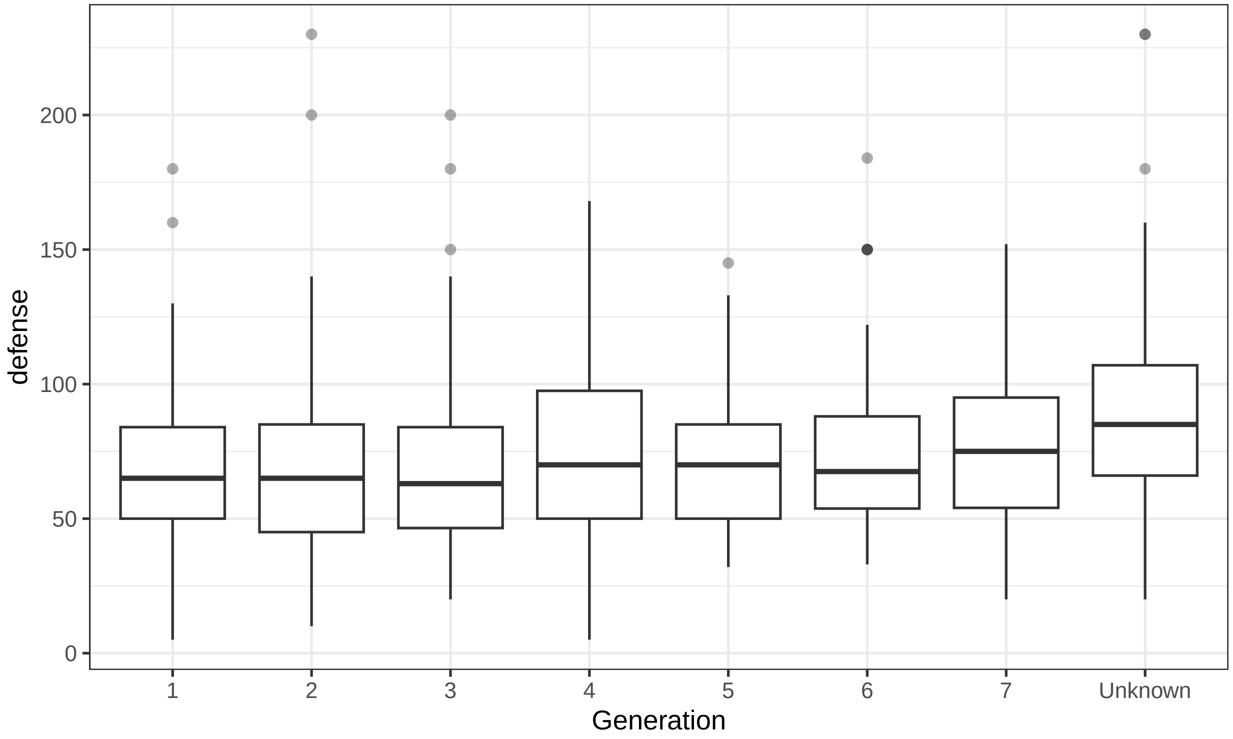
```
## Saved: figures/box_attack_by_generation.png
```

Distribution of attack by generation (including Unknown)

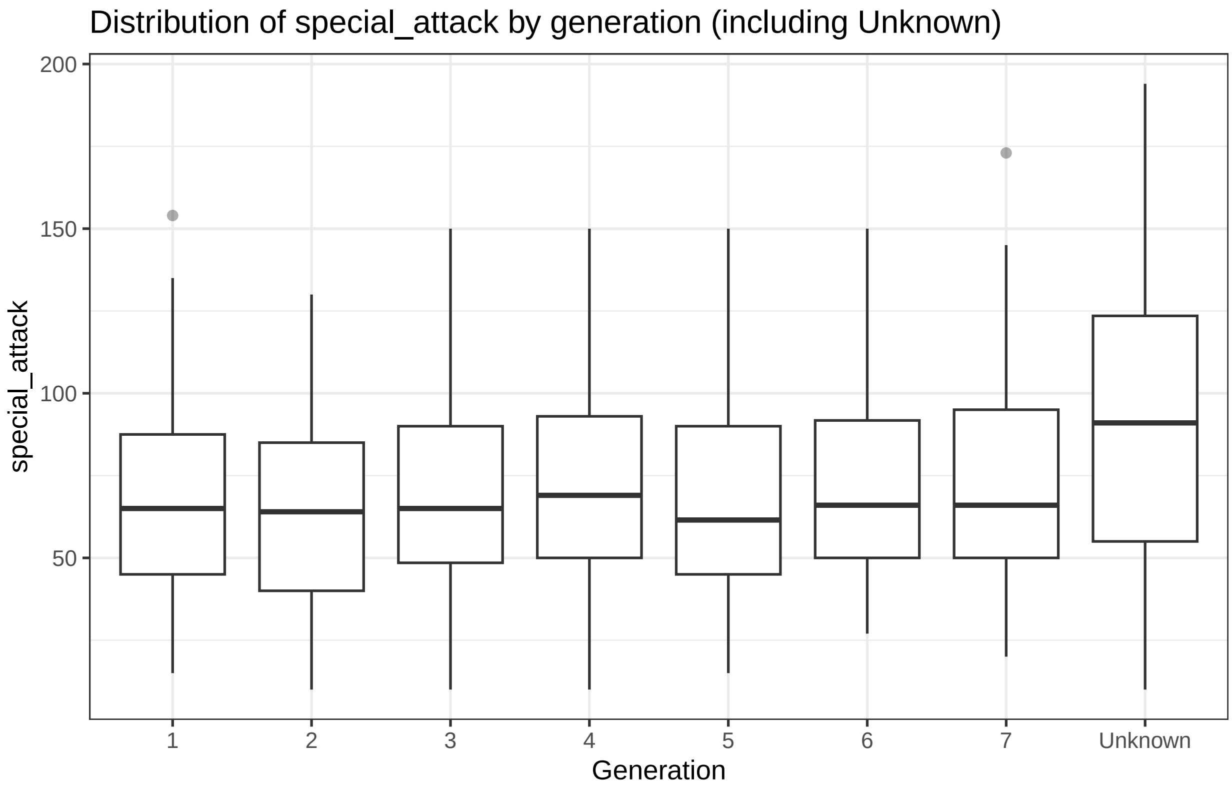


```
## Saved: figures/box_defense_by_generation.png
```

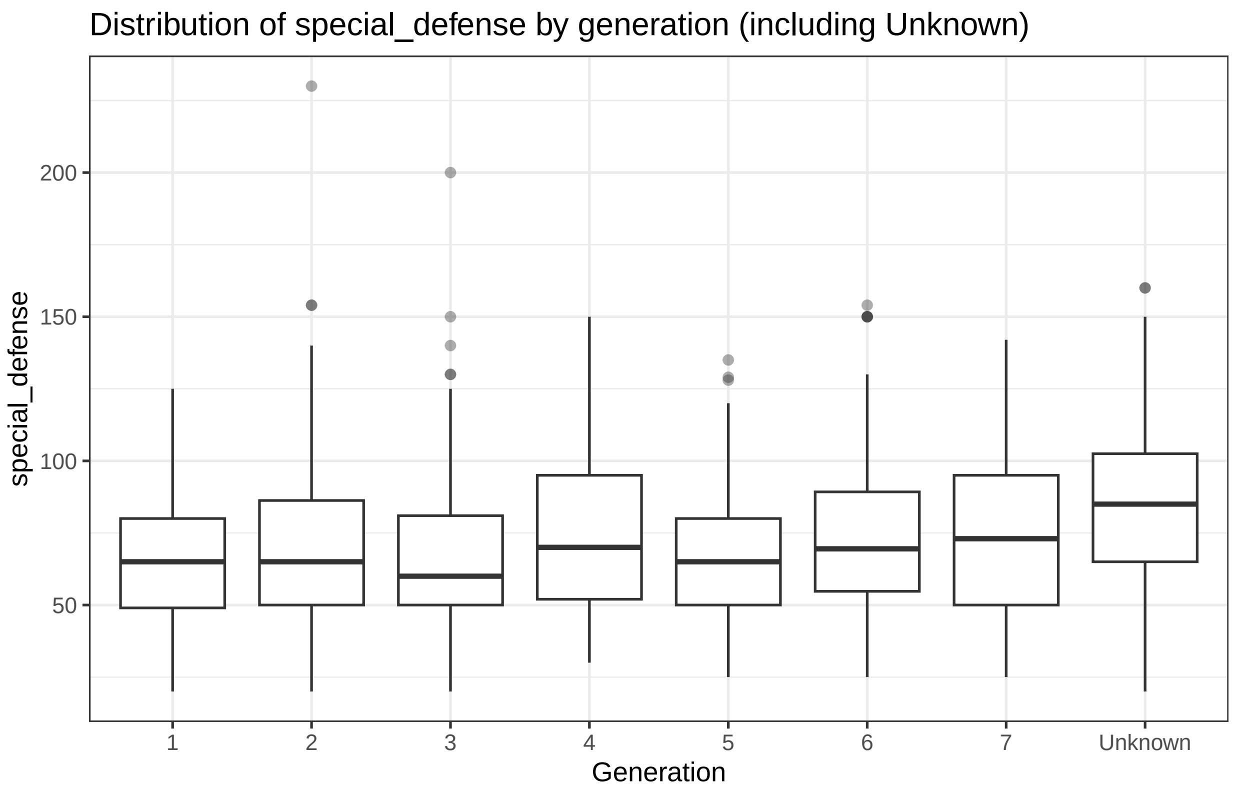
Distribution of defense by generation (including Unknown)



```
## Saved: figures/box_special_attack_by_generation.png
```

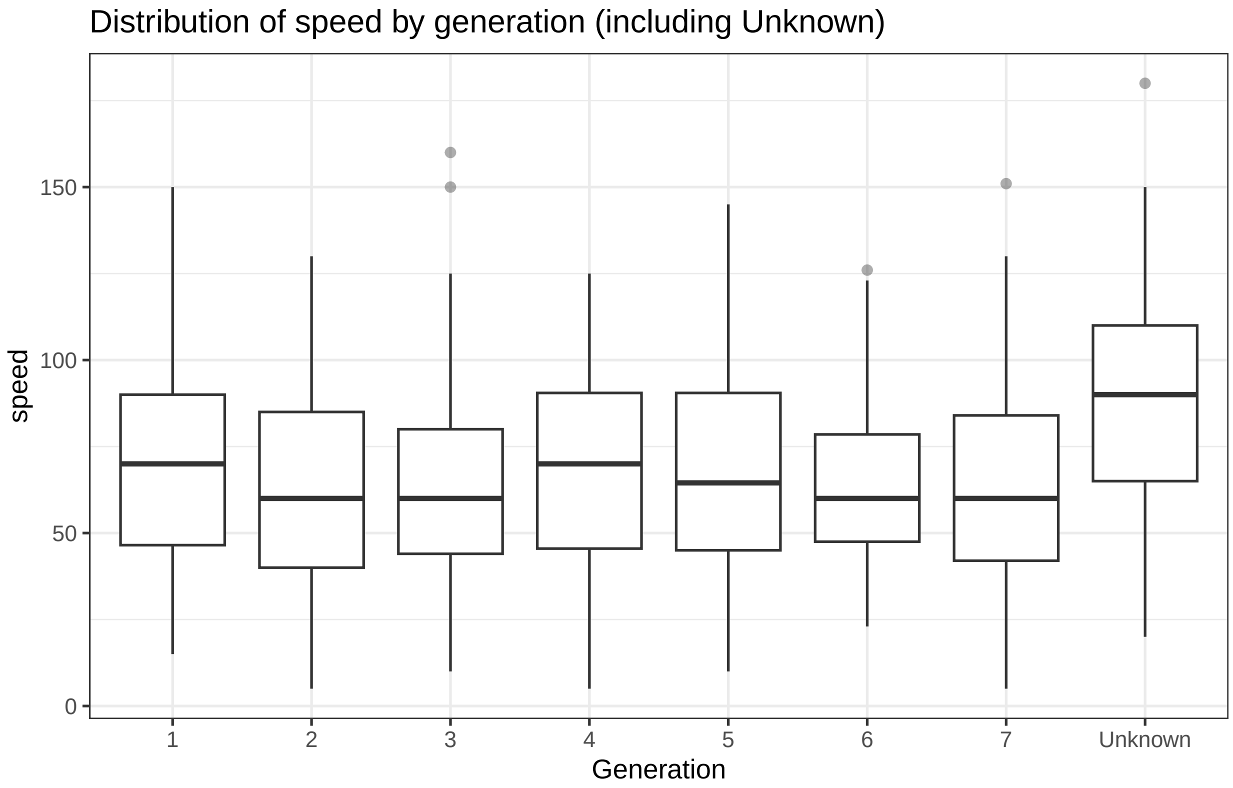


```
## Saved: figures/box_special_defense_by_generation.png
```



Distribution of special_defense by generation (including Unknown)

```
## Saved: figures/box_speed_by_generation.png
```



Distribution of speed by generation (including Unknown)

```r
# Distributions of abilities by type1
#| label: stats-by-type1
#| message: false
#| warning: false

# Ensure type_1 is a factor; order by frequency for nicer plots
pokemon_df <- pokemon_df |>
  mutate(
    type_1_f = fct_infreq(as.factor(type_1))
  )

# Summary table by primary type
type1_stats <- pokemon_df |>
  group_by(type_1_f) |>
  summarise(
    n = n(),
    across(
      all_of(stat_vars),
      list(mean = ~mean(., na.rm = TRUE),
           sd   = ~sd(.,   na.rm = TRUE)),
      .names = "{.col}_{.fn}"
    )
  ) |>
  arrange(desc(n))

readr::write_csv(type1_stats,
                 "tables/type1_stats_all_stats.csv")

knitr::kable(
  type1_stats,
  digits = 1,
  caption = "Means and standard deviations of stats by primary type (type_1)."
)
```

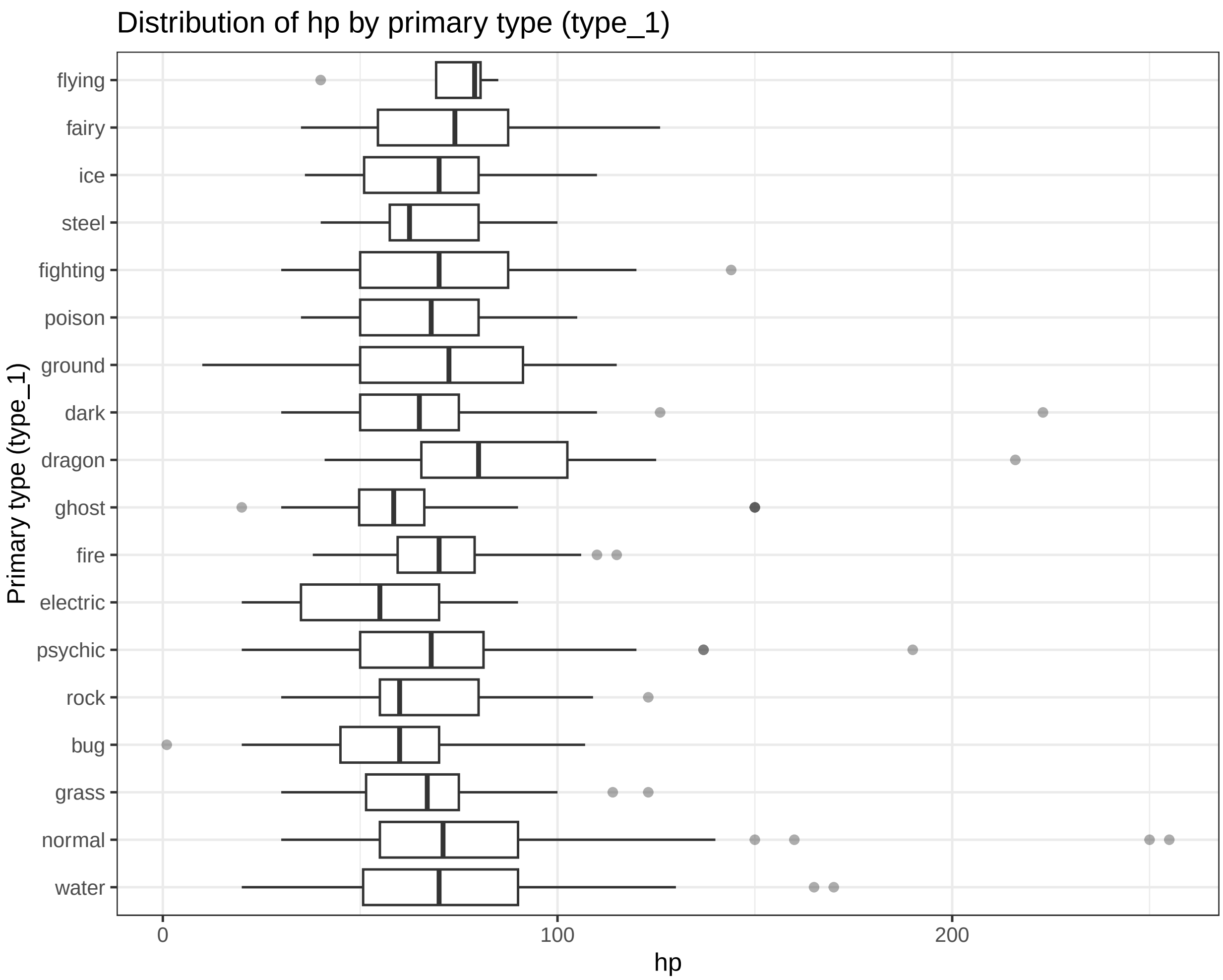
Table 7: Means and standard deviations of stats by primary type (type_1).

type_1_f	n	hp_mean	hp_sd	attack_mean	attack_sd	defense_mean	defense_sd	special_attack_mean	special_attack_sd	special_defense_mean	special_defense_sd	speed_mean	speed_sd
water	126	71.1	26.4	74.7	29.0	73.5	28.1	75.6	30.3	72.4	29.6	65.7	24.8
normal	111	77.3	34.8	75.9	30.0	60.5	23.5	57.5	25.1	64.4	25.2	70.3	27.7
grass	84	66.7	18.9	75.2	29.3	71.7	25.0	76.2	27.0	70.7	22.1	60.3	27.9
bug	79	57.7	17.3	72.5	37.2	72.1	34.1	57.7	31.0	64.4	31.0	63.3	34.0
rock	65	65.3	19.4	90.0	31.9	94.3	34.4	66.2	28.0	75.3	30.2	64.3	33.2
psychic	64	72.6	29.8	72.5	42.0	69.9	29.1	98.0	39.1	87.0	31.1	80.9	36.3
electric	61	55.8	18.3	68.1	22.4	61.2	23.7	83.0	32.6	69.1	21.4	87.2	24.0
fire	59	69.7	18.8	84.3	27.5	69.3	25.0	87.6	29.0	71.9	22.0	73.5	24.8
ghost	40	64.2	28.7	76.3	28.5	82.0	29.6	77.3	30.8	78.7	25.5	65.7	29.1
dragon	39	84.5	32.0	108.7	32.4	89.3	25.0	93.4	39.8	88.3	28.4	82.9	22.8
dark	37	69.7	33.2	84.7	26.3	67.6	24.5	71.4	32.8	67.8	24.1	76.6	26.8
ground	36	71.6	27.5	96.2	32.2	82.6	33.5	55.3	26.8	62.9	20.7	64.6	27.9
poison	35	67.2	19.6	73.5	19.8	69.3	24.4	62.5	21.7	65.9	23.6	64.2	25.7
fighting	31	71.6	25.7	99.1	28.0	67.2	18.2	53.8	27.3	64.9	21.9	67.6	26.9

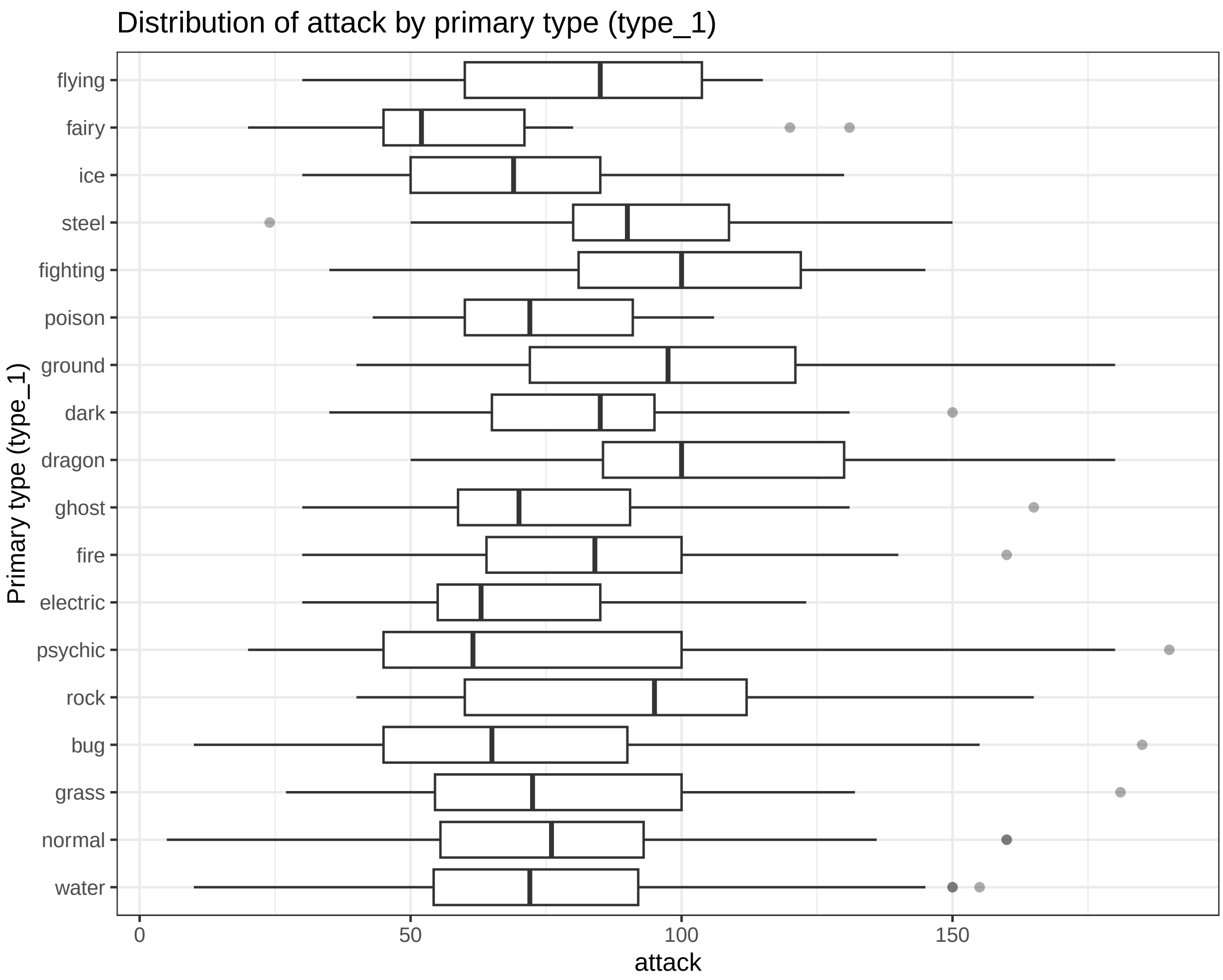
type_1_f	n	hp_mean	hp_sd	attack_mean	attack_sd	defense_mean	defense_sd	special_attack_mean	special_attack_sd	special_defense_mean	special_defense_sd	speed_mean	speed_sd
steel	30	67.3	16.6	93.1	28.8	124.8	42.7	73.0	34.3	83.6	29.0	56.1	24.6
ice	29	70.5	20.8	73.1	27.3	73.4	32.5	72.3	29.1	74.7	35.0	64.6	24.2
fairy	19	72.9	22.9	61.2	28.1	67.1	18.7	81.2	28.9	88.3	30.2	53.6	26.6
flying	4	70.8	20.7	78.8	37.5	66.2	21.4	94.2	34.8	72.5	22.2	102.5	32.1

```r
# Boxplots for each stat by primary type (flipped for readability)
for (s in stat_vars) {
  p <- ggplot(pokemon_df,
              aes(x = type_1_f, y = .data[[s]])) +
    geom_boxplot(outlier.alpha = 0.4) +
    coord_flip() +
    labs(
      title = paste0("Distribution of ", s, " by primary type (type_1)"),
      x = "Primary type (type_1)",
      y = s
    ) +
    theme_bw()

  file_name <- paste0("figures/box_", s, "_by_type1.png")
  ggsave(file_name, p, width = 7.5, height = 6, dpi = 150)

  cat("Saved:", file_name, "\n")
  print(p)  # show in the knitted report
}
```

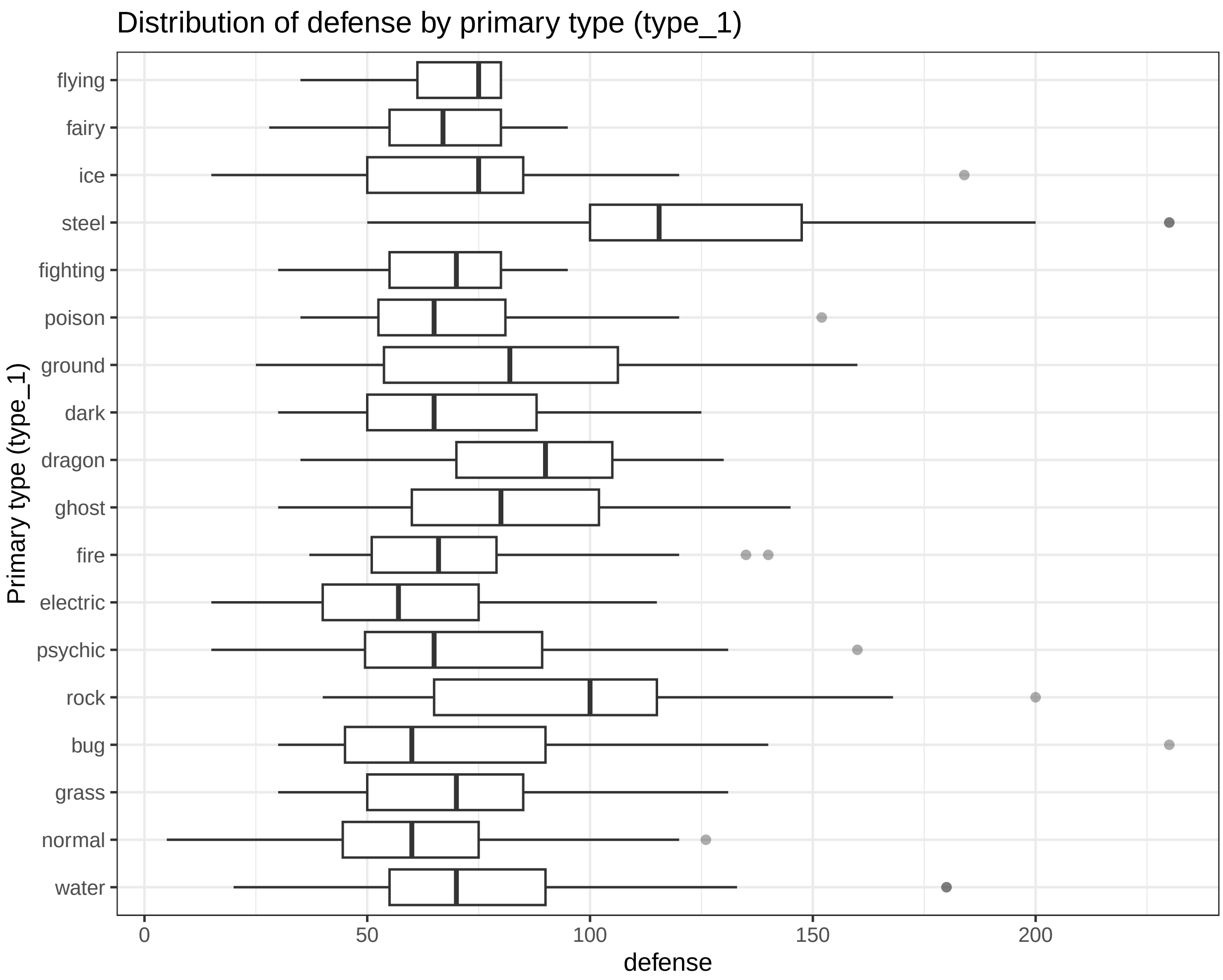
```
## Saved: figures/box_hp_by_type1.png
```



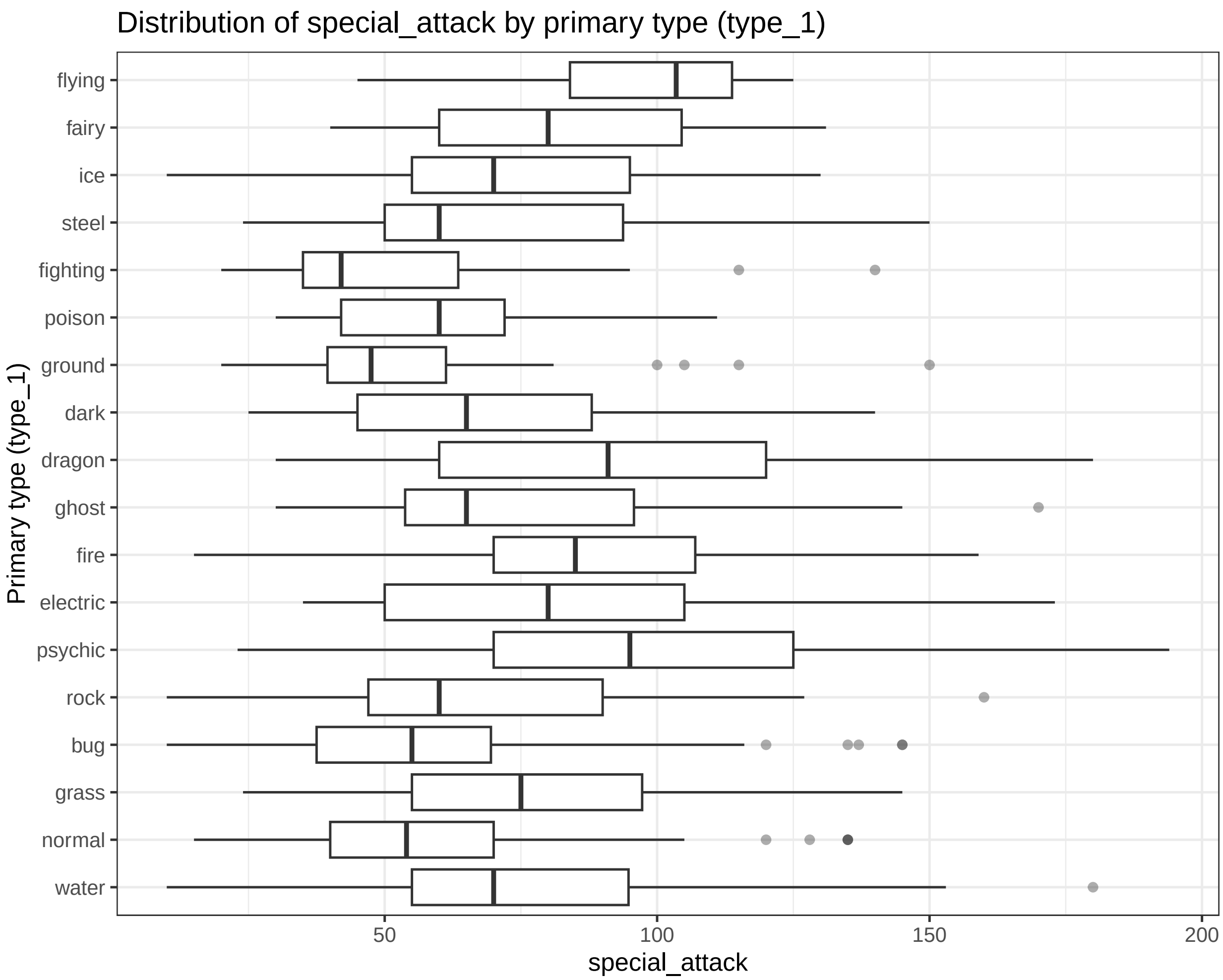
```
## Saved: figures/box_attack_by_type1.png
```



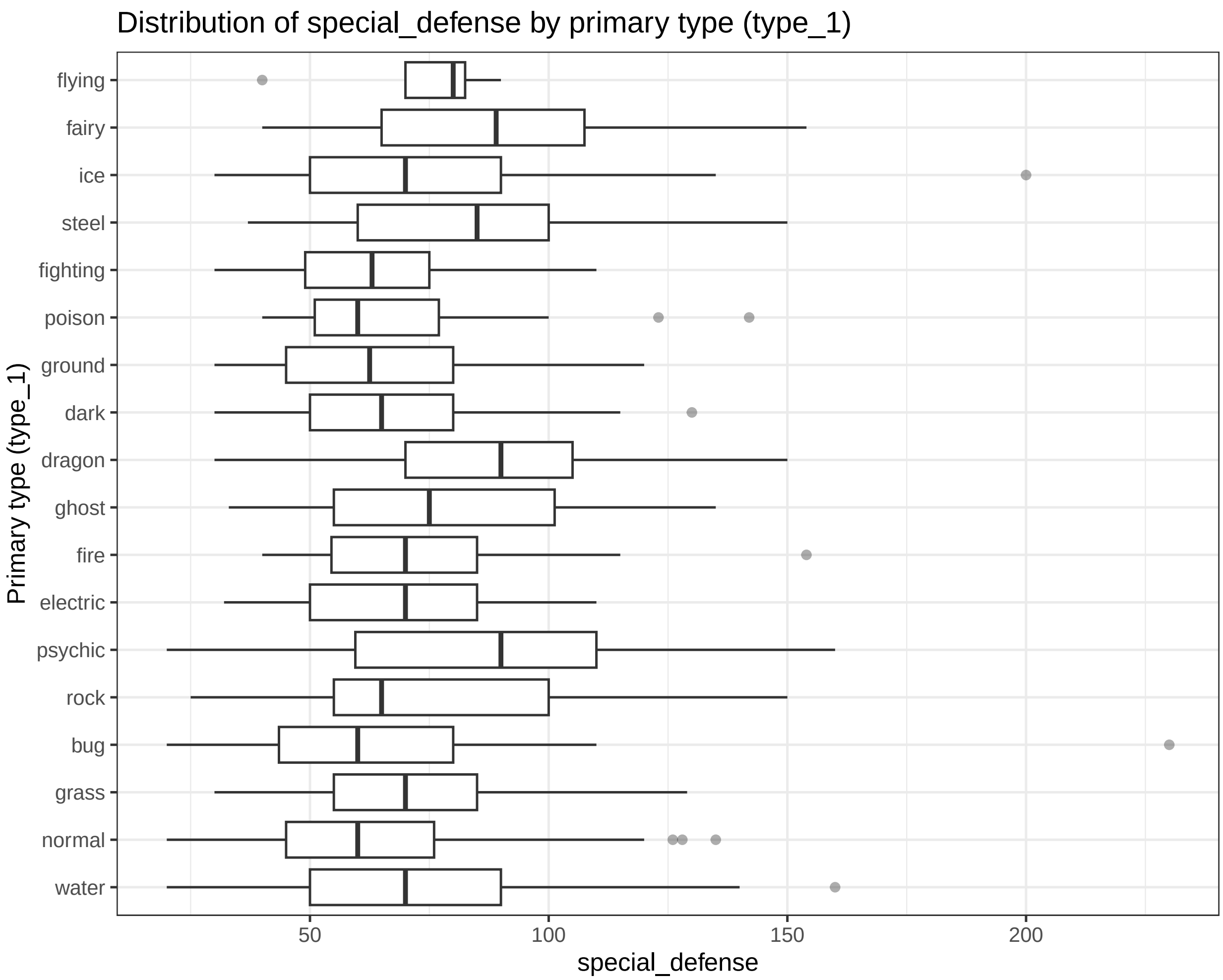
```
## Saved: figures/box_defense_by_type1.png
```



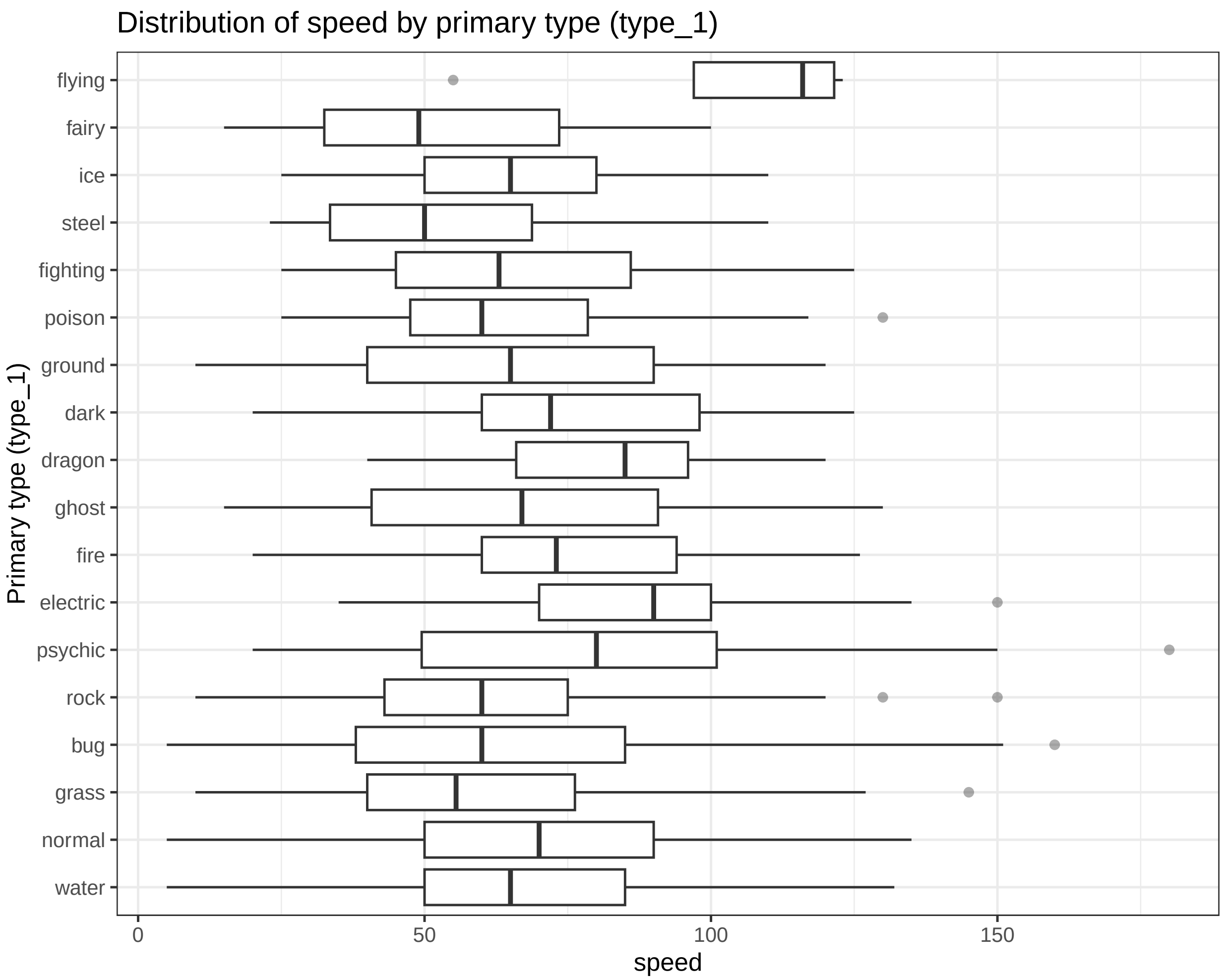
```
## Saved: figures/box_special_attack_by_type1.png
```



```
## Saved: figures/box_special_defense_by_type1.png
```



```
## Saved: figures/box_speed_by_type1.png
```


Distribution of speed by primary type (type_1)
Primary type (type_1)
flying
fairy
ice
steel
fighting
poison
ground
dark
dragon
ghost
fire
electric
psychic
rock
bug
grass
normal
water
0
50
100
150
speed