AE 01: Visualizing the prognosticators

Suggested answers

APPLICATION EXERCISE ANSWERS

MODIFIED
September 13, 2024

Important

These are suggested answers. This document should be used as reference only, it's not designed to be an exhaustive key.

Note

Below are the contents of the YAML header. You normally do not see this in the rendered file - we include it so you can reproduce the figure dimensions in the document.

```yaml
---
title: "AE 01: Visualizing the prognosticators"
subtitle: "Suggested answers"
execute:
  fig-width: 8
  fig-height: 4
  warning: false
---
```

For all analyses, we'll use the **tidyverse** packages.

```
library(tidyverse)
library(scales)
```

Data: The prognosticators

The dataset we will visualize is called `seers`.[1] It contains summary statistics for all known Groundhog Day forecasters. [2] Let's `glimpse()` at it.

[1] I would prefer `prognosticators`, but I had way too many typos preparing these materials to make you all use it.

[2] Source: Countdown to Groundhog Day. Application exercise inspired by Groundhogs Do Not Make Good Meteorologists originally published on FiveThirtyEight.

```
# import data using readr::read_csv()
seers <- read_csv("data/prognosticators-sum-stats.csv")

glimpse(seers)
```

```
Rows: 154
Columns: 18
$ name                  <chr> "Allen McButterpants", "Arboretum Annie", "Babyl…
$ forecaster_type       <chr> "Groundhog", "Groundhog", "Groundhog Mascot", "S…
$ forecaster_simple     <chr> "Groundhog", "Groundhog", "Groundhog Mascot", "O…
$ alive                 <lgl> TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, TRUE,…
$ climate_region        <chr> "Northeast", "South", "Northeast", "Northeast", …
$ town                  <chr> "Hampton Bays", "Dallas", "Babylon", "Bridgeport…
$ state                 <chr> "NY", "TX", "NY", "CT", "TX", "OH", "TX", "TX", …
$ preds_n               <dbl> 2, 3, 1, 13, 14, 9, 8, 1, 1, 12, 2, 4, 13, 10, 1…
$ preds_long_winter     <dbl> 0, 1, 0, 1, 3, 4, 5, 1, 1, 6, 2, 2, 8, 9, 0, 1, …
$ preds_long_winter_pct <dbl> 0.00000000, 0.33333333, 0.00000000, 0.07692308, …
$ preds_correct         <dbl> 2, 2, 1, 10, 10, 6, 4, 0, 0, 5, 0, 2, 5, 2, 0, 2…
$ preds_rate            <dbl> 1.0000000, 0.6666667, 1.0000000, 0.7692308, 0.71…
$ temp_mean             <dbl> 33.70000, 50.18333, 35.05000, 31.43462, 51.57143…
$ temp_hist             <dbl> 30.31167, 51.32333, 30.47667, 29.63667, 50.99310…
$ temp_sd               <dbl> 4.154767, 3.908807, 4.154767, 4.154767, 3.908807…
$ precip_mean           <dbl> 3.007500, 2.768333, 3.620000, 3.059231, 2.577500…
$ precip_hist           <dbl> 3.0251667, 2.5588889, 3.0760000, 3.0700000, 2.56…
$ precip_sd             <dbl> 0.9715631, 0.8999887, 0.9715631, 0.9715631, 0.89…
```

The variables are:

- `name` - name of the prognosticator
- `forecaster_type` - what kind of animal or thing is the prognosticator?
- `forecaster_simple` - a simplified version that lumps together the least-frequently appearing types of prognosticators
- `alive` - is the prognosticator an animate (alive) being?[3]
- `climate_region` - the NOAA climate region in which the prognosticator is located.
- `town` - self-explanatory
- `state` - state (or territory) where prognosticator is located
- `preds_n` - number of predictions in the database
- `preds_long_winter` - number of predictions for a “Late Winter” (as opposed to “Early Spring”)
- `preds_long_winter_pct` - percentage of predictions for a “Late Winter”
- `preds_correct` - number of correct predictions[4]
- `preds_rate` - proportion of predictions that are correct
- `temp_mean` - average temperature (in Fahrenheit) in February and March in the climate region across all prognostication years
- `temp_hist` - average of the rolling 15-year historic average temperature in February and March across all prognostication years

- `temp_sd` - standard deviation of average February and March temperatures across all prognostication years
- `precip_mean` - average amount of precipitation in February and March across all prognostication years (measured in rainfall inches)
- `precip_hist` average of the rolling 15-year historic average precipitation in February and March across all prognostication years
- `precip_sd` - standard deviation of average February and March precipitation across all prognostication years

[3] Prognosticators labeled as Animatronic/Puppet/Statue/Stuffed/Taxidermied are classified as not alive.

[4] We adopt the same definition as FiveThirtyEight. An “Early Spring” is defined as any year in which the average temperature in either February or March was higher than the historic average. A “Late Winter” was when the average temperature in both months was lower than or the same as the historical average.

Visualizing prediction success rate - Demo

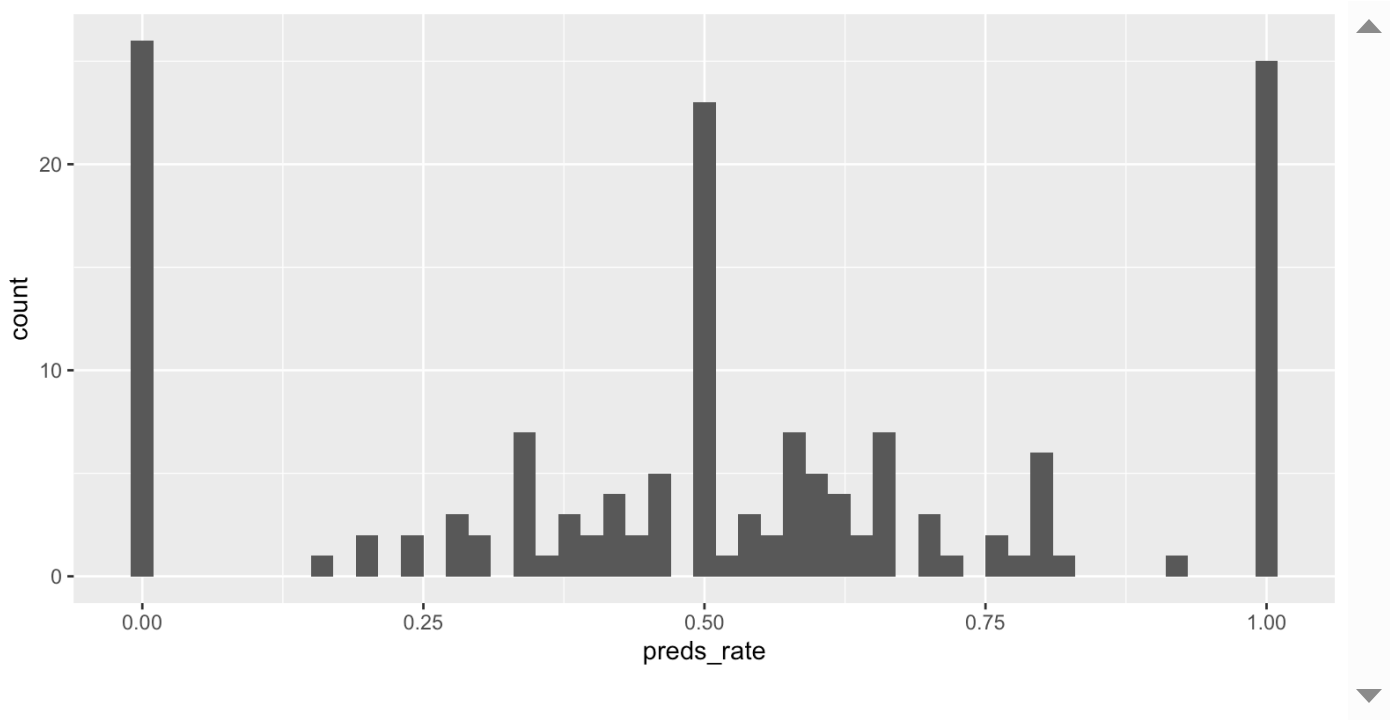
Single variable

> **Note**
>
> Analyzing the a single variable is called **univariate** analysis.

Create visualizations of the distribution of `preds_rate` for the prognosticators.

1. Make a histogram. Set an appropriate binwidth.

```
ggplot(data = seers, mapping = aes(x = preds_rate)) +
  geom_histogram(binwidth = 0.02)
```



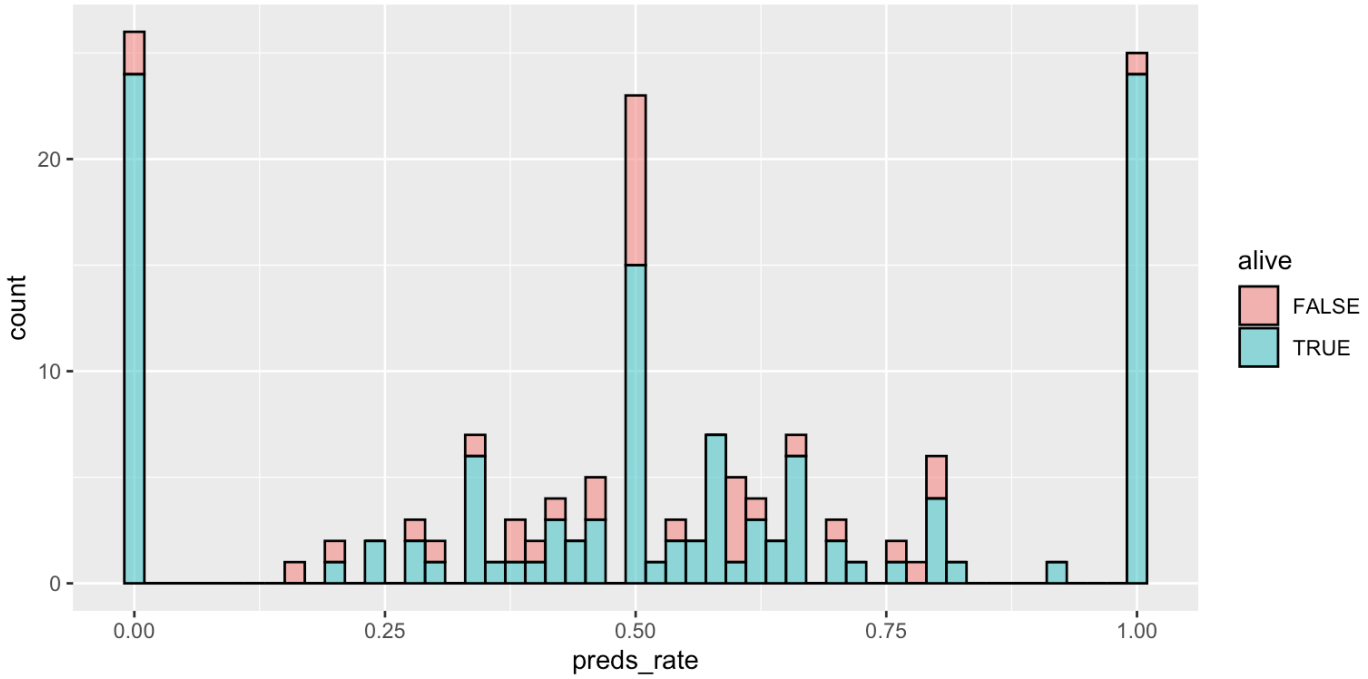
Two variables - Your turn

> **Note**
>
> Analyzing the relationship between two variables is called **bivariate** analysis.

Create visualizations of the distribution of `preds_rate` by `alive` (whether or not the prognosticator is alive).

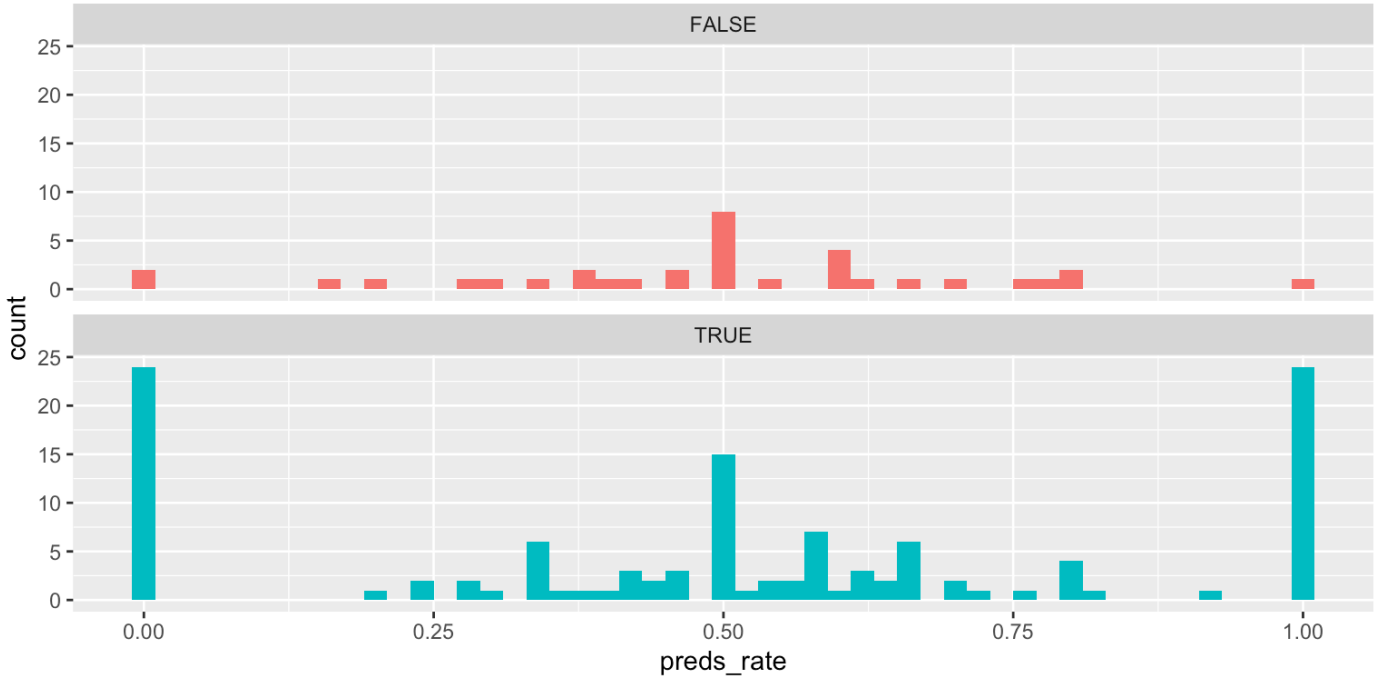
2. Make a single histogram. Set an appropriate binwidth.

```
ggplot(
  data = seers,
  mapping = aes(x = preds_rate, fill = alive)
) +
  geom_histogram(binwidth = 0.02, alpha = 0.5, color = "black")
```



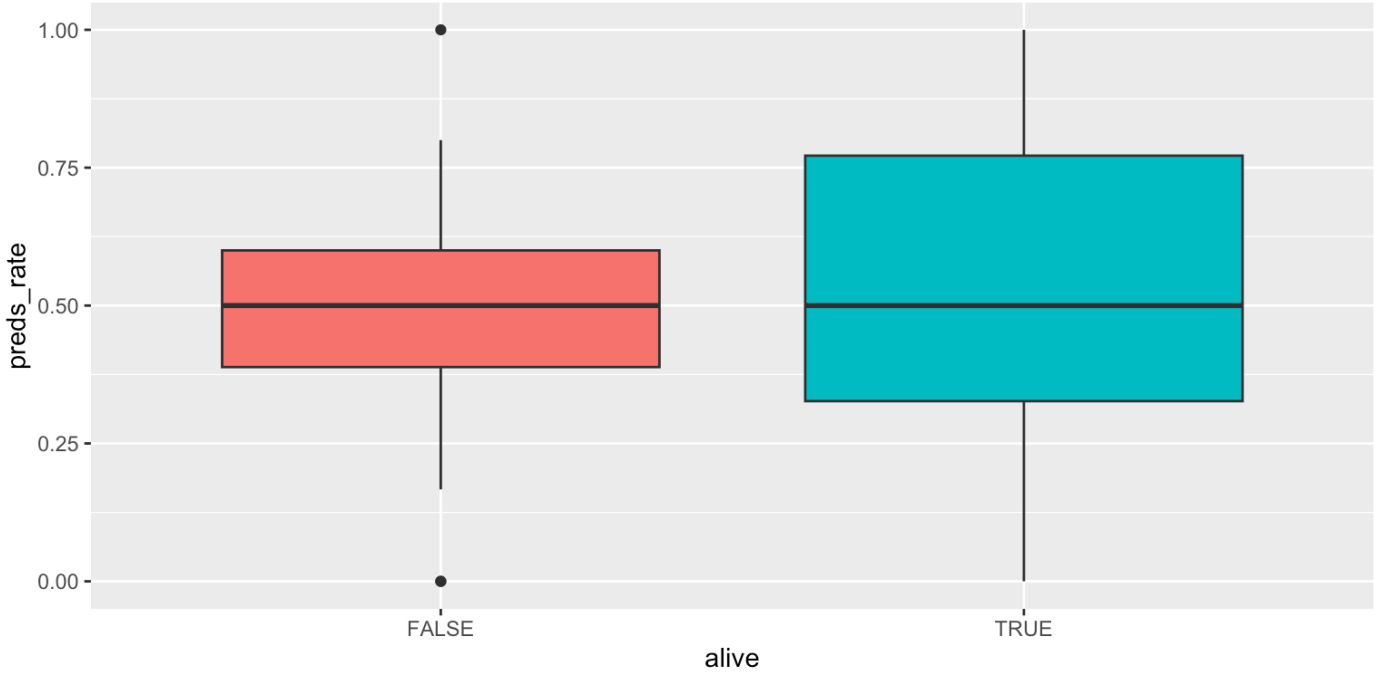
3. Use multiple histograms via faceting, one for each type. Set an appropriate binwidth, add color as you see fit, and turn off legends if not needed.

```
ggplot(
  data = seers,
  mapping = aes(x = preds_rate, fill = alive)
) +
  geom_histogram(binwidth = 0.02, show.legend = FALSE) +
  facet_wrap(vars(alive), ncol = 1)
```



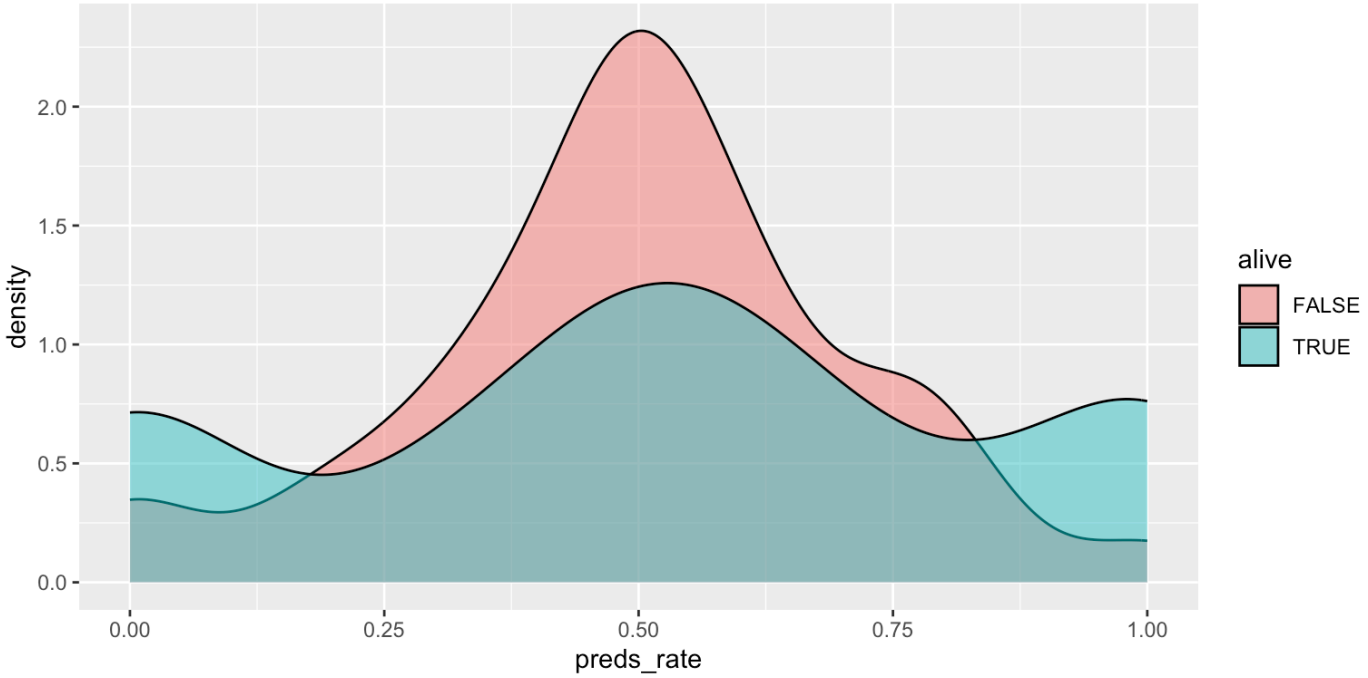
4. Use side-by-side box plots. Add color as you see fit and turn off legends if not needed.

```
ggplot(
  data = seers,
  mapping = aes(x = alive, y = preds_rate, fill = alive)
) +
  geom_boxplot(show.legend = FALSE)
```



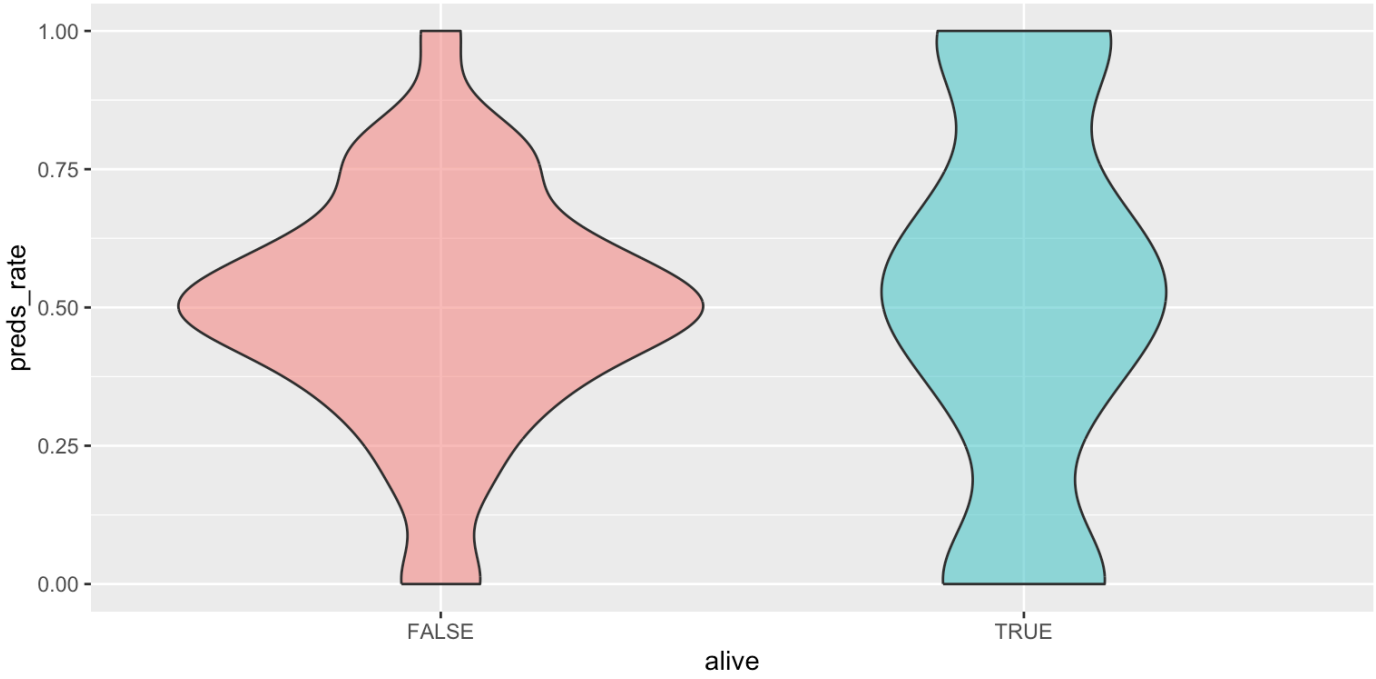
5. Use a density plot. Add color as you see fit.

```
ggplot(
  data = seers,
  mapping = aes(x = preds_rate, fill = alive)
) +
  geom_density(alpha = 0.5)
```



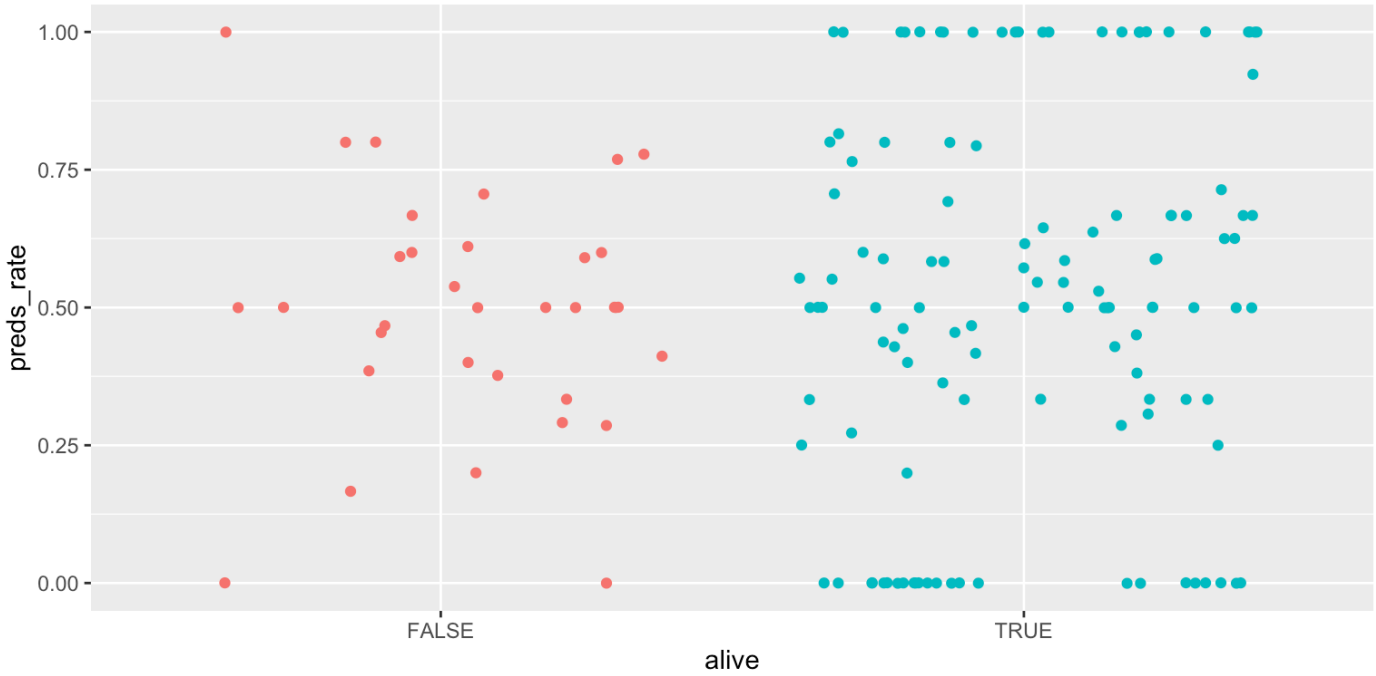
6. Use a violin plot. Add color as you see fit and turn off legends if not needed.

```
ggplot(
  data = seers,
  mapping = aes(x = alive, y = preds_rate, fill = alive)
) +
  geom_violin(alpha = 0.5, show.legend = FALSE)
```



7. Make a jittered scatter plot. Add color as you see fit and turn off legends if not needed.

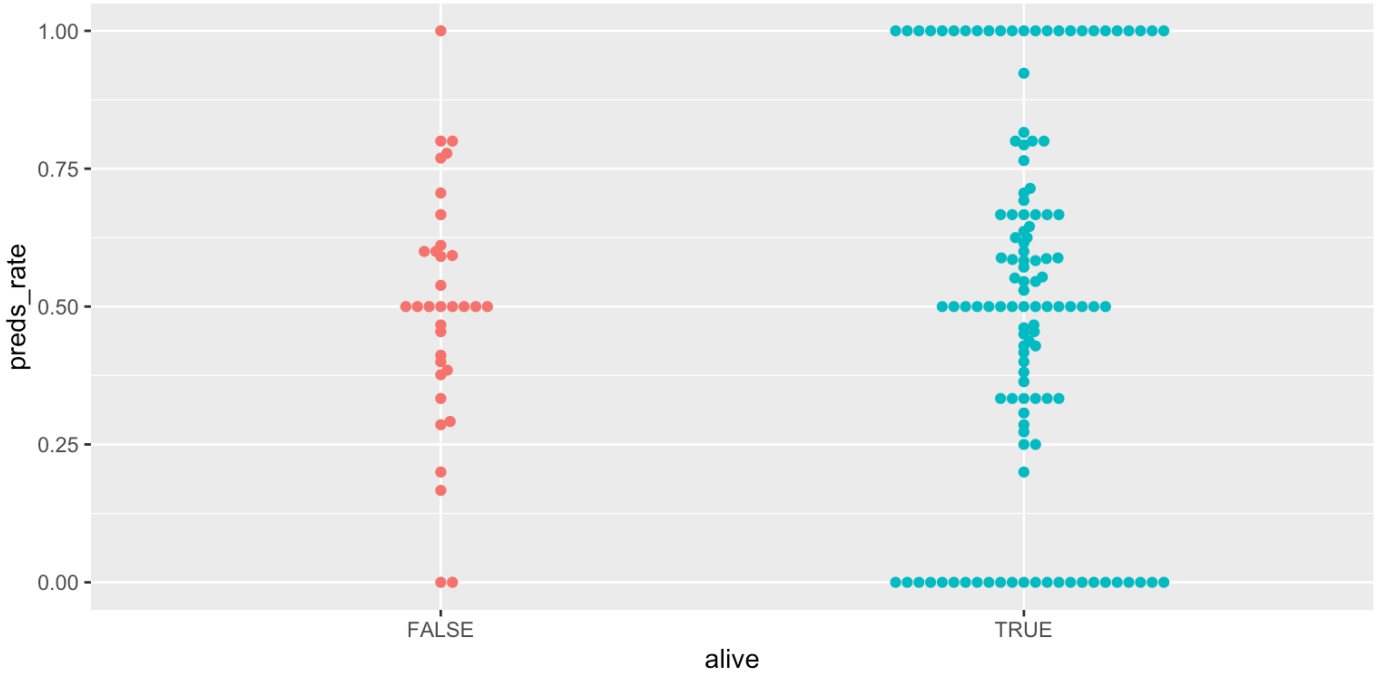
```
ggplot(
  data = seers,
  mapping = aes(x = alive, y = preds_rate, color = alive)
) +
  geom_jitter(show.legend = FALSE)
```



8. Use beeswarm plots. Add color as you see fit and turn off legends if not needed.

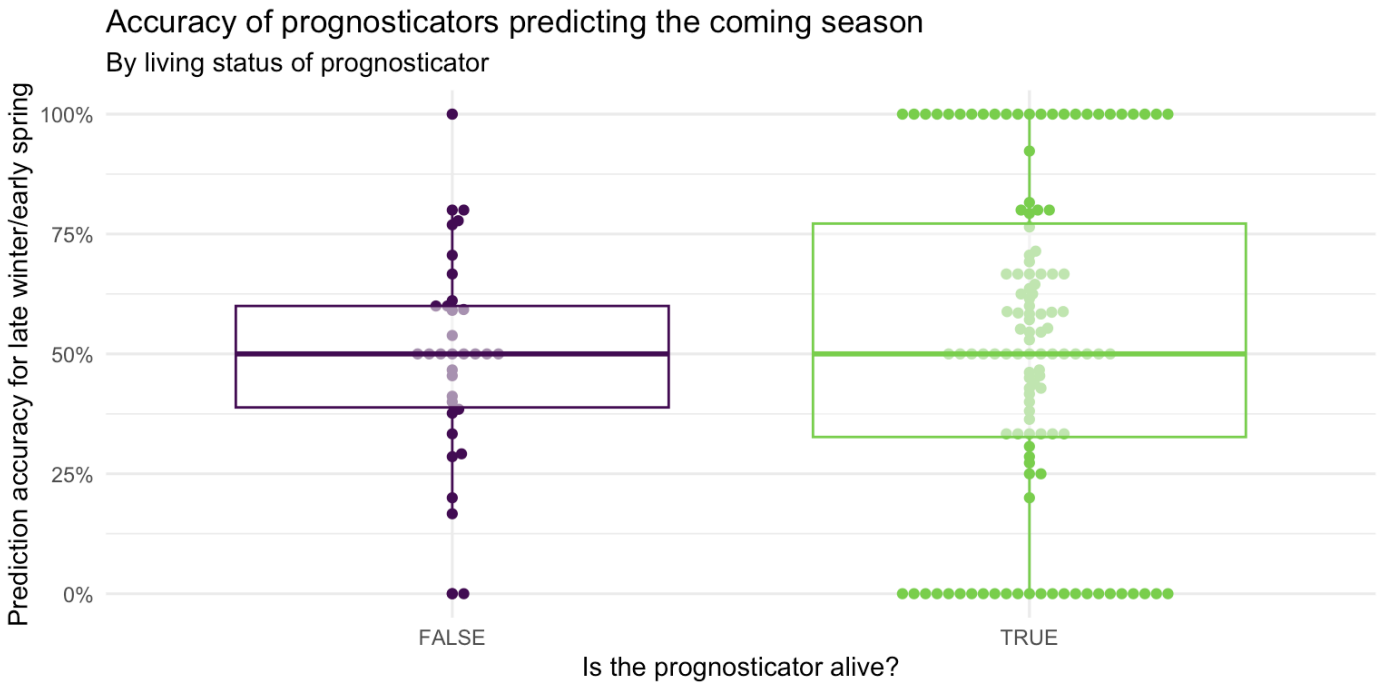
```
library(ggbeeswarm)

ggplot(
  data = seers,
  mapping = aes(x = alive, y = preds_rate, color = alive)
) +
  geom_beeswarm(show.legend = FALSE)
```



9. **Demonstration**: Use multiple geoms on a single plot. Be deliberate about the order of plotting. Change the theme and the color scale of the plot. Finally, add informative labels.

```
ggplot(
  data = seers,
  mapping = aes(x = alive, y = preds_rate, color = alive)
) +
  geom_beeswarm(show.legend = FALSE) +
  geom_boxplot(show.legend = FALSE, alpha = 0.5) +
  scale_color_viridis_d(option = "D", end = 0.8) +
  scale_y_continuous(labels = label_percent()) +
  theme_minimal() +
  labs(
    x = "Is the prognosticator alive?",
    y = "Prediction accuracy for late winter/early spring",
    title = "Accuracy of prognosticators predicting the coming season",
    subtitle = "By living status of prognosticator"
  )
```

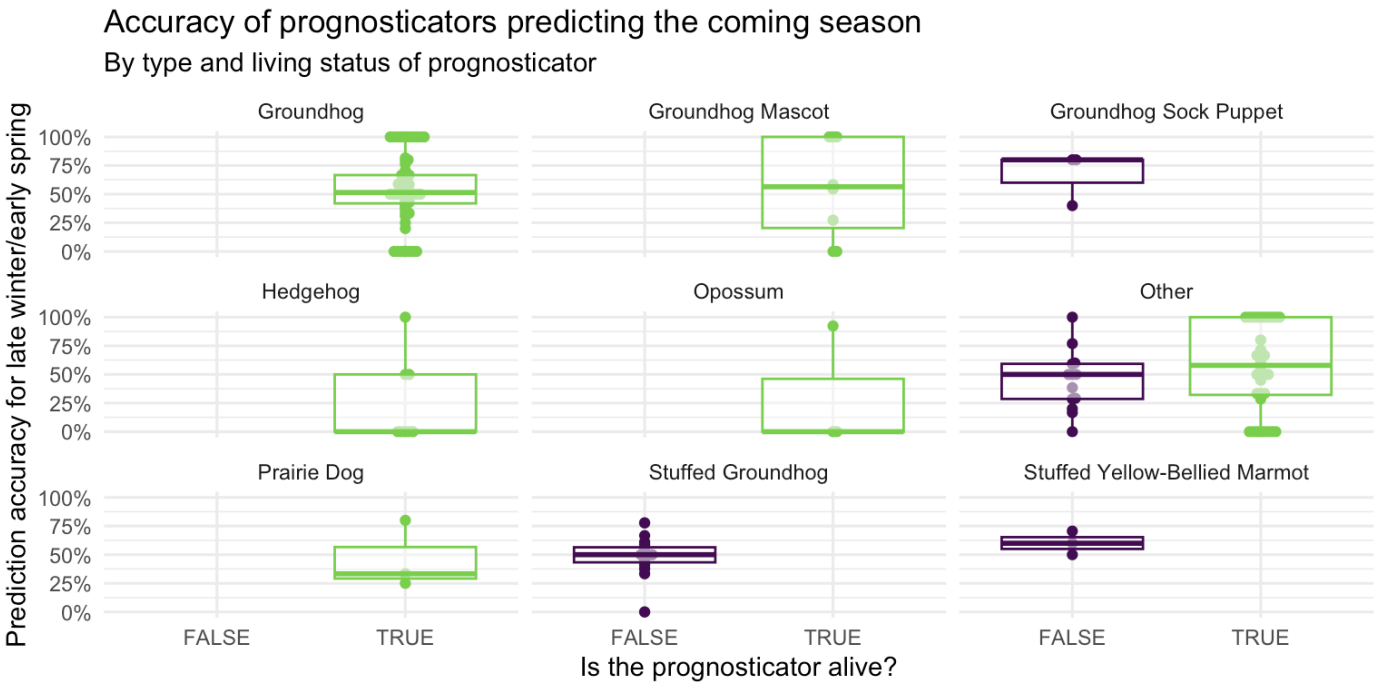


Multiple variables - Demo

> **Note**
>
> Analyzing the relationship between three or more variables is called **multivariate** analysis.

10. Facet the plot you created in the previous exercise by `forecaster_simple`. Adjust labels accordingly.

```
ggplot(
  data = seers,
  mapping = aes(x = alive, y = preds_rate, color = alive)
) +
  geom_beeswarm(show.legend = FALSE) +
  geom_boxplot(show.legend = FALSE, alpha = 0.5) +
  scale_color_viridis_d(option = "D", end = 0.8) +
  scale_y_continuous(labels = label_percent()) +
  facet_wrap(vars(forecaster_simple)) +
  theme_minimal() +
  labs(
    x = "Is the prognosticator alive?",
    y = "Prediction accuracy for late winter/early spring",
    title = "Accuracy of prognosticators predicting the coming season",
    subtitle = "By type and living status of prognosticator"
  )
```

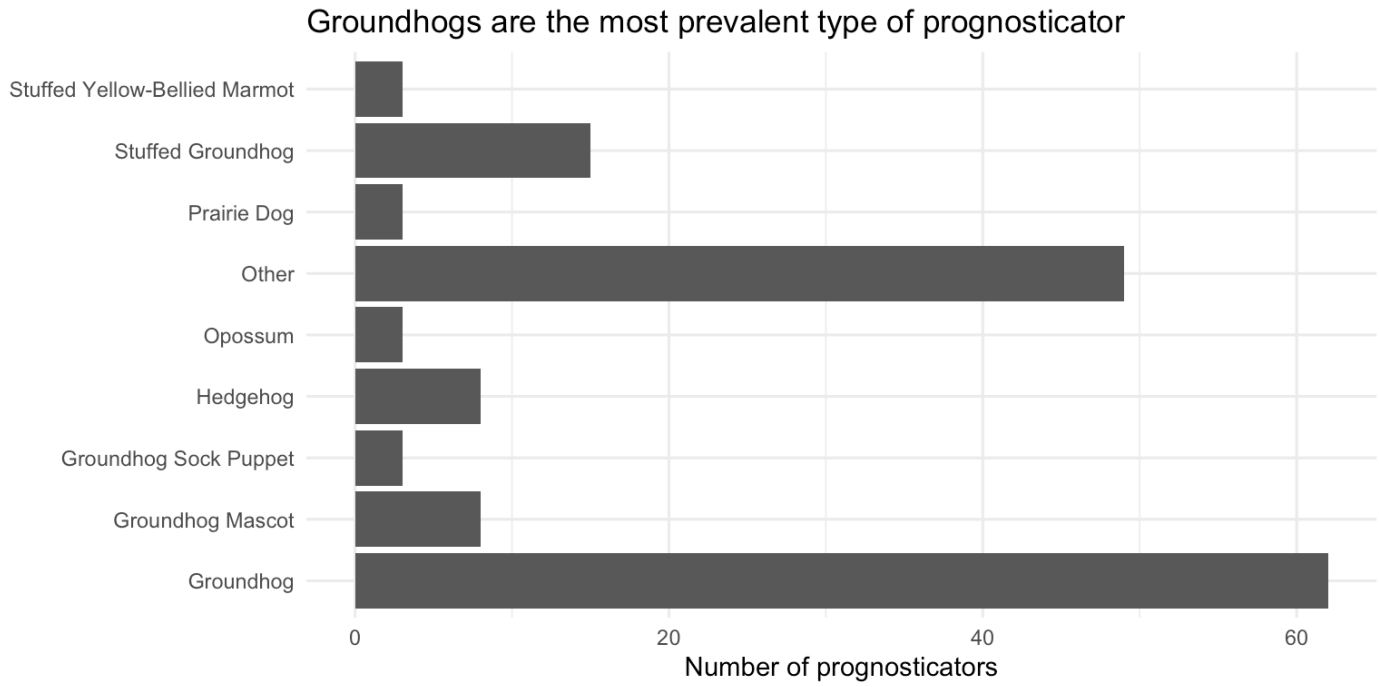


Before you continue, let's turn off all warnings the code chunks generate and resize all figures. We'll do this by editing the YAML.

Visualizing other variables - Your turn!

11. Pick a single categorical variable from the data set and make a bar plot of its distribution.

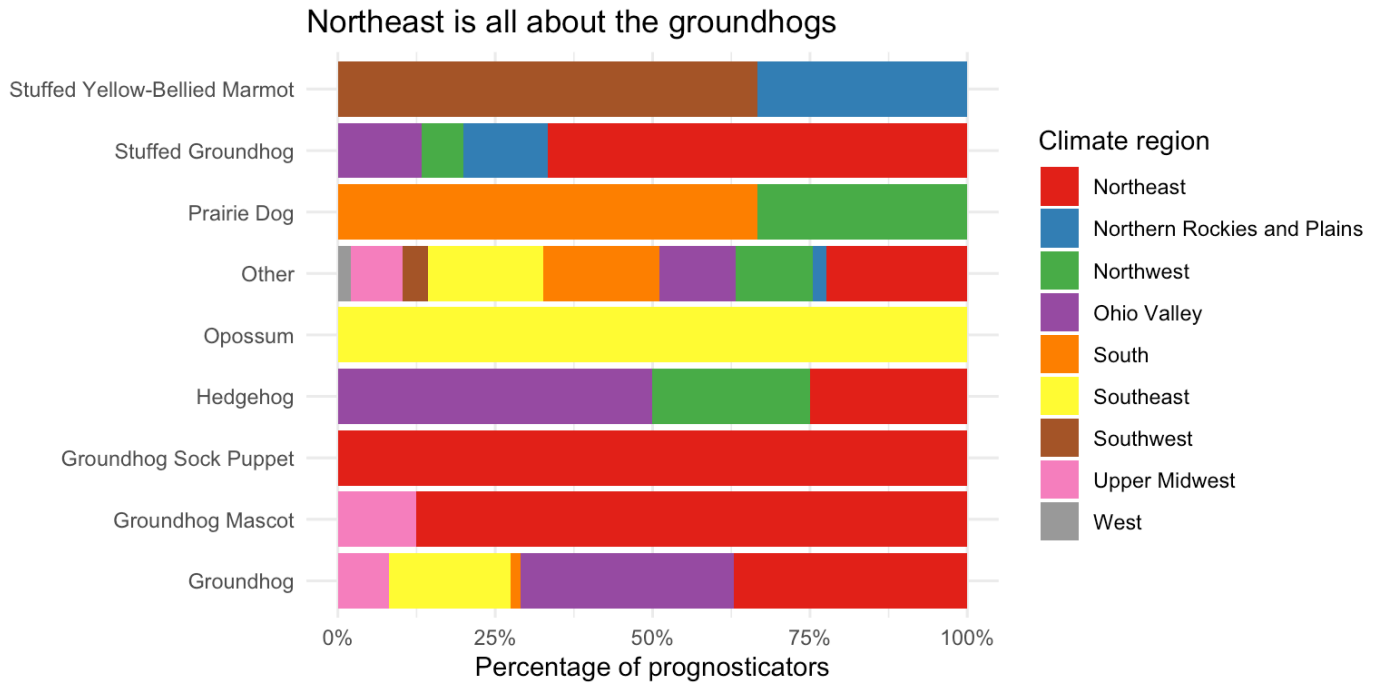
```
# frequency count of forecaster_simple
ggplot(data = seers, mapping = aes(y = forecaster_simple)) +
  geom_bar() +
  theme_minimal() +
  labs(
    title = "Groundhogs are the most prevalent type of prognosticator",
    x = "Number of prognosticators",
    y = NULL
  )
```



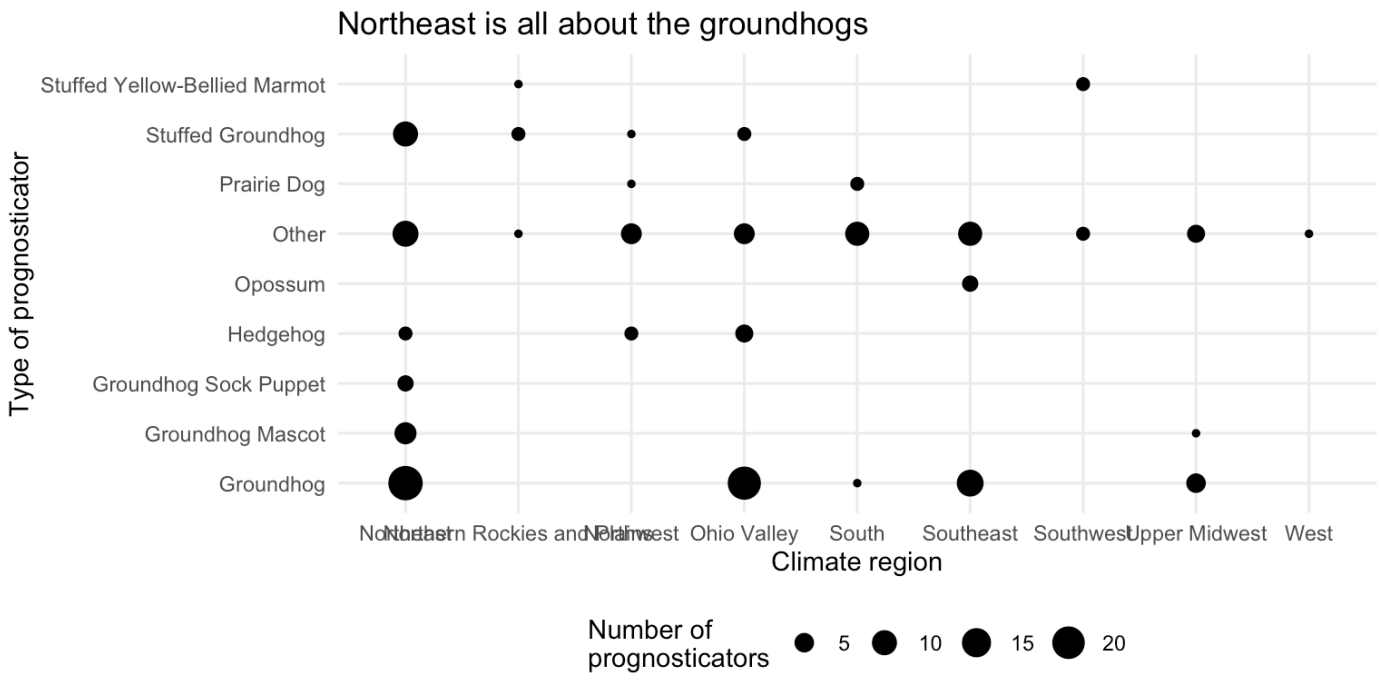
The y-axis is sorted alphabetically which doesn't make a lot of sense, but fixing it requires some data wrangling (which we will learn next week!). Overall groundhogs are the most prevalent type of prognosticator.

12. Pick two categorical variables and make a visualization to visualize the relationship between the two variables. Along with your code and output, provide an interpretation of the visualization.

```
# forecaster_simple vs. climate_region as a relative frequency bar chart
ggplot(data = seers, mapping = aes(y = forecaster_simple, fill = climate_region)) +
  # position = "fill" makes the bars the same height
  geom_bar(position = "fill") +
  # percentage labeling
  scale_x_continuous(labels = label_percent()) +
  # use better color palette
  scale_fill_brewer(type = "qual", palette = "Set1") +
  theme_minimal() +
  labs(
    title = "Northeast is all about the groundhogs",
    x = "Percentage of prognosticators",
    y = NULL,
    fill = "Climate region"
  )
```



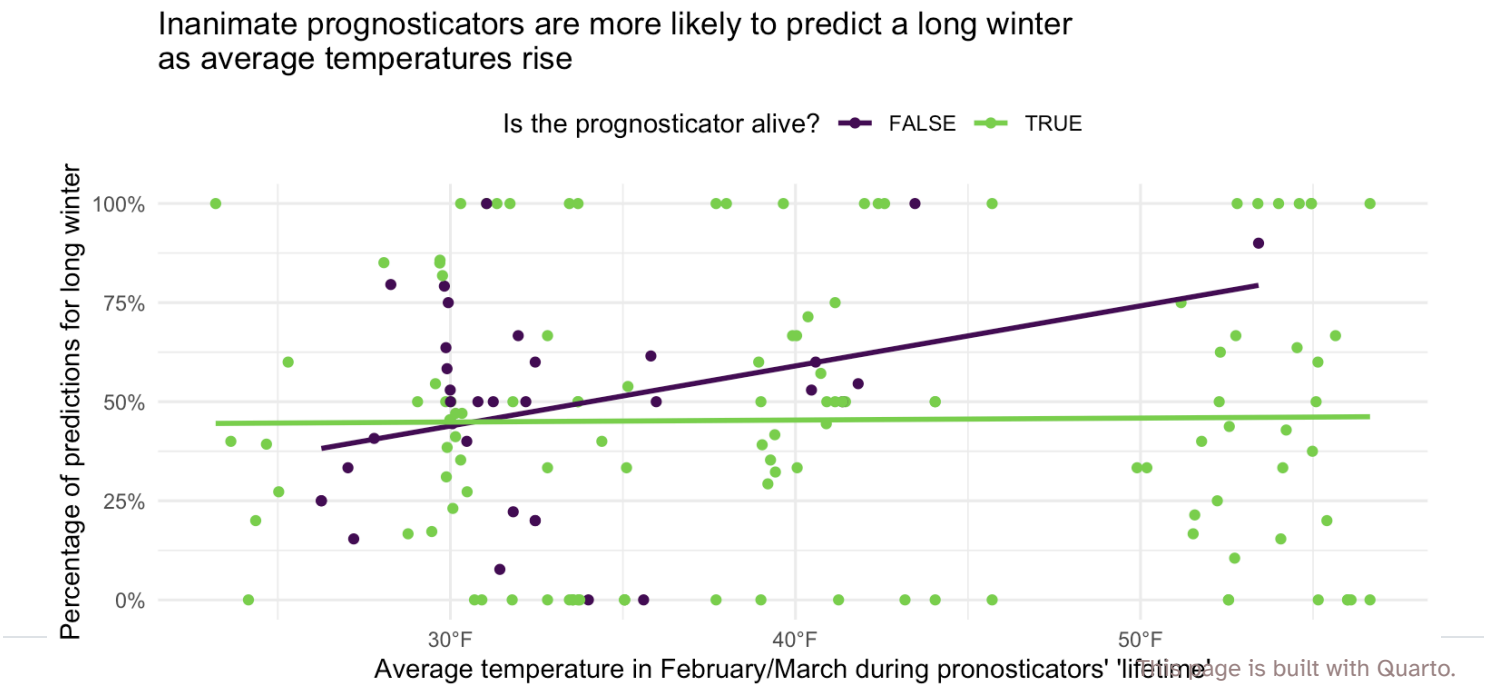

```
# now as a count plot
ggplot(data = seers, mapping = aes(x = climate_region, y = forecaster_simple)) +
  geom_count() +
  labs(
    title = "Northeast is all about the groundhogs",
    x = "Climate region",
    y = "Type of prognosticator",
    size = "Number of\nprognosticators"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



It's rather hard to visualize two categorical variables at once. The first plot shows the relative frequency of each type of prognosticator by climate region. The second plot shows the count of each type of prognosticator by climate region. Both plots show that the Northeast has a high proportion of groundhogs as prognosticators. Unfortunately we end up with overlapping labels on the second plot, while with the first graph we cannot tell the overall number of prognosticators by each type (everything is scaled to 100%).

13. Make another plot that uses at least three variables. At least one should be numeric and at least one categorical. In 1-2 sentences, describe what the plot shows about the relationships between the variables you plotted. Don't forget to label your code chunk.

```
ggplot(data = seers, mapping = aes(x = temp_mean, y = preds_long_winter_pct,
                                   color = alive)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_continuous(
    labels = label_number(suffix = "\u00b0F")
  ) +
  scale_color_viridis_d(option = "D", end = 0.8) +
  scale_y_continuous(labels = label_percent()) +
  labs(
    title = "Inanimate prognosticators are more likely to predict a long winter\nas average
         temperatures rise",
    x = "Average temperature in February/March during pronosticators' 'lifetime'",
    y = "Percentage of predictions for long winter",
    color = "Is the prognosticator alive?"
  ) +
  theme_minimal() +
  theme(legend.position = "top")
```



As average temperatures in February and March rise, inanimate prognosticators are associated with a higher likelihood of predicting a long winter. For living prognosticators, the relationship is less clear. The smoothing line is virtually flat, suggesting there is no association between average winter temperatures and whether or not the prognosticator predicts a long winter or early spring.

Session information