PS: 12 NAME:

## R Packages II

Each of the following questions refer to the `kknn` package (https://github.com/KlausVigo/kknn) that we'll be using for predictive modeling.

1. What is the author's one sentence description of this package?

2. Which data sets are stored in this package?

3. Unlike most packages, you've used, you can't just type the name of a data set to see it (try it with `glass`). Instead you first have to use `data(glass)` to specifically load that data set. What is the name this property of a package? Does it exist in the DESCRIPTION file?

4. What is the name of the file in the package that is edited to created the help files for these data sets?

---

Return to the tiny package that you made called `greetr`.

5. Add the following tibble to the your package as a data set called `greetings`.

```r
library(tibble)
greetings <- tibble(
  greeting = c("Hello", "□□", "Hola", "Hei", "السلام عليكم"),
  language = c("English", "Chinese", "Spanish", "Norwegian", "Urdu")
)
```

6. Document this data set using `roxygen2` so that when a user types `?greetings` at the console they see a description of the data set, its format (the columns and their values), and its source (you can cite STAT 133). Please transcribe your documentation below.

7. Add documentation for your previous function called `hello()`. Be sure to include a description, the arguments, the output, and an example of its use. Please transcribe your documentation below.

8. One last item: bump your version number up by a small increment (say from 0.1.0 to 0.1.1) in the `DESCRIPTION` file. Write that single new line from your DESCRIPTION file below.

## Prediction II

Start by piecing together the code cells from the slides that use `tidymodels` to predict the `price` of diamonds using the value of `carat`. Put them into an .r script or Quarto document.

The following exercises have you investigate the different components of the code and make variations.

9. Run all of the code and inspect the final data frame. As measured by RMSE (Root Mean Squared Error), which of the two models has better performance on the test set?

10. Modify the code to add a third model to the mix: a KNN model with 50 instead of 15 neighbors. As measured by RSME, how does its predictive performance on the test set compare to the original two models?

11. What is the purpose of the `initial_split()` function? What is the type of the R object that it creates? What is the first element in that object and what does it represent?

12. What does the `training` function do? What is the type of the R object that it creates?

13. To understand the effect of the different steps of the data preparation recipe, make a duplicate of that pipeline and change the name of the result to `transformed`. Add two more functions to the pipeline: `prep()` and `bake()`[1]. Each beyond the piped recipe object, each one takes a data frame as an argument: use the training data.

    Compare `transformed_train` to `diamonds_train`. What changes have been made to the data as a result of this recipe?

[1] The authors of `recipes` went all in on the baking metaphor.

14. Add two additional predictors to the recipe: `depth` and `table`[2]. What do these variables represent about diamonds?

15. Rerun your original three models with this modified recipe. How does the predictive performance of each model on the test set compare?

[2] In R's formula interface, you add additional predictors adding them to the right side: .e.g `y ~ x1 + x2`.