Midterm 1

Stat 220: Spring 2024

2024-04-24

Name:

Total Points: 100

Q1 Consider the following objects and determine what each of the code snippets evaluates to. Briefly explain your answers.

```
x <- -2:2
y <- c(FALSE, factor(c("b", "NA")), 3, NA)
z <- list(z1 = x,
          z2 = y,
          z3 = c("Carleton", "College"),
          z4 = matrix(1:9, nrow = 3))
```

(a)

```
stringr::str_flatten(z[[3]], collapse = " ")
```

(b)

```
z[[2]] - z[[1]]
```

Q2 You are provided with a dataset containing simulated data representing properties' prices, square footage, and construction years. The dataset is designed to reflect realistic property characteristics:

- Square Footage (`sqft`): This variable is generated using the `runif` function to produce 100 random numbers following a uniform distribution, specifically ranging from 1000 to 2000 square feet.
- Construction Year (`year`): This variable lists the years properties were constructed, sampled uniformly from 1970 to 2020.
- Price (`price`): The price of each property is generated using the `rnorm` function, which produces 100 random numbers following a normal distribution. The mean price is set with a formula based on both the year of construction and the square footage, calculated as $\$10,000$ plus $\$10,000$ for every year past 1970 and an additional \$500 for each square foot over 1000. The standard deviation is set at $\$50,000$, indicating variability in property prices due to factors not included in our simple model.

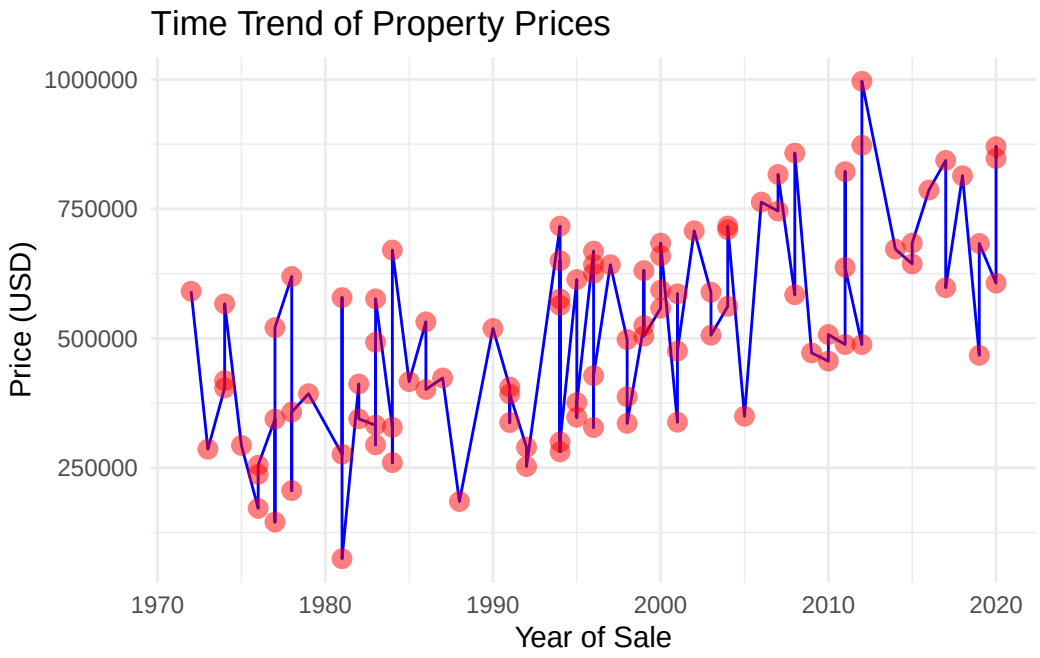
```
set.seed(123) # for reproducibility
data <- tibble(
  year = sample(1970:2020, 100, replace = TRUE),
  sqft = runif(100, 1000, 2000),
  price = rnorm(100,
                mean = 10000 + (year - 1970) * 10000 + (sqft- 1000) * 500,
                sd = 50000)
)

head(data) # first 6 rows of the dataset
```

```
# A tibble: 6 x 3
   year  sqft   price
  <int> <dbl>   <dbl>
1  2000 1320. 558093.
2  1984 1308. 328140.
3  2020 1220. 606597.
4  1983 1369. 332325.
5  1972 1984. 590940.
6  2011 1154. 488081.
```

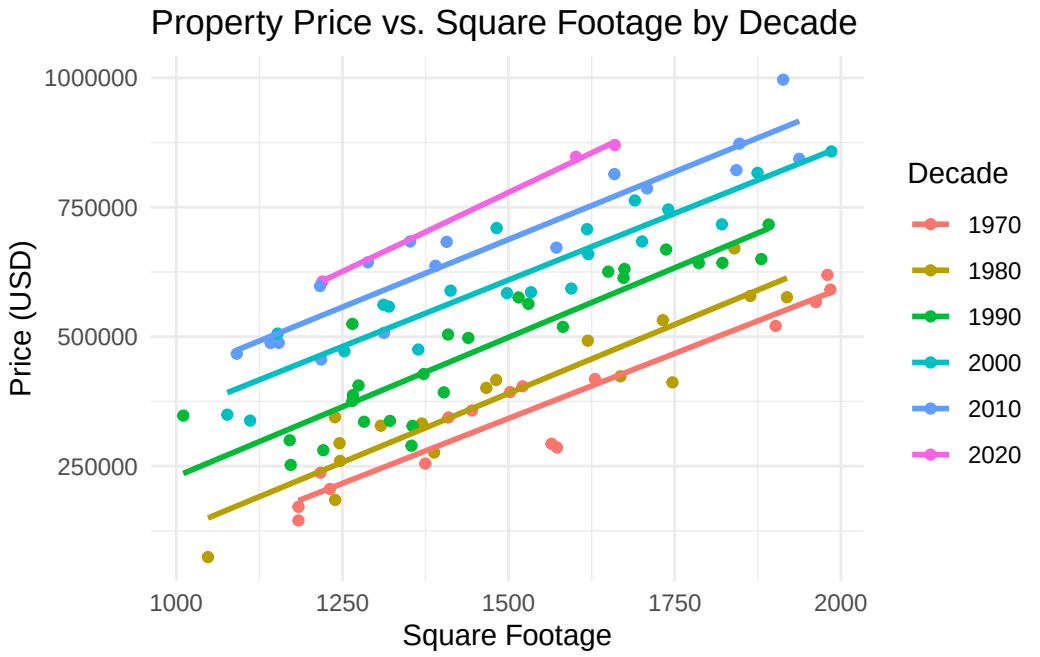
(a) Write code to add a new column called `decade` to this dataset that records the decade of each year, make it a factor, and store it in `data_new`. A decade should be represented as the first year of the decade (e.g., 1980 for any year from 1980 to 1989).

(b) Fill in the missing parts of a `ggplot2` code snippet to create a time trend plot that displays the trend of property prices over the years. (Refer to the plot on the next page!)

```
ggplot(data_new, aes(x = _____________, y = ____________)) +
  geom_line(_________________________) +
  geom_point(___________________, _____________________, ___________________) +
  labs(title = "_________________________________",
       x = "__________________",
       y = "__________________") +
  theme_minimal()
```



(c) Write code to create a scatter plot using `ggplot2` that displays property prices against square footage. Fit a linear model to these data points and color-code the points based on the decade of sale. Fill in the details in the provided code snippet.



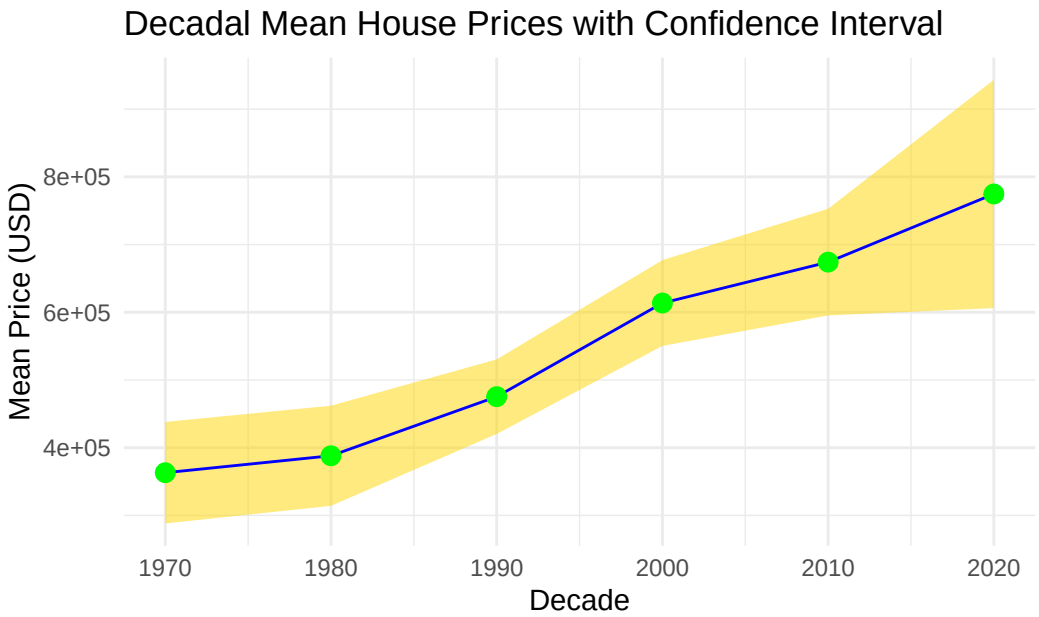
```
ggplot(data_new, aes(x = _____________, y = _____________, color = __________)) +
  geom_point() +
  geom_smooth(method = "_____________", se = __________) +
  labs(title = "_____________________________________",
       x = "_________________________",
       y = "_________________________",
       color = "_____________") +
  theme_minimal()
```

(d) Calculate the mean price and standard error of the price for each decade using the `data_new` data object from part (a) and store it in `stats_by_decade`, which contains information about property prices and the decade of sale. Remember, the formula for standard error is $\mathrm{SE} = \frac{\mathrm{SD}}{\sqrt{n}}$, where n is the number observations in each group.

(e) Complete the following code to produce a ribbon plot using the `stats_by_decade` from part (d), which includes the mean and standard error of property prices per decade. Ensure that you transform the decade variable to numeric for appropriate plotting, include a ribbon representing the confidence interval within ±2 standard errors of the mean, and add points and lines connecting the means at each decade. (Refer to the plot on the next page!)

```
stats_by_decade %>%
  _________(decade = ___________________(decade)) %>%
  ggplot(_________(x = _____________, y = _____________)) +
  geom_ribbon(______(ymin = _________________, ymax = _________________),
              fill = "_____", alpha = _______) +
  geom_line(_____________________) +
  geom_point(__________________, _________________) +
  labs(title = "_____________________________________________",
```

```
      x = "_____________________________",
      y = "_____________________________",
      caption = "_____________________________________________________________") +
  theme_minimal()
```



Q3 Miscellaneous

(a) Given a vector of date strings formatted as `ddmmyyyy` below, convert these strings into UTC date-time objects using the `lubridate` package, and then calculate the duration between each consecutive date in days.

```
library(lubridate)
dates <- c("01012023", "15032024")
```

(b) You are provided with a factor variable `experience` with levels representing professional experience: "Entry", "Mid", "Senior". Reverse the order of these levels to reflect descending order of experience and store it inside `experience_reversed`.

```
library(forcats)
experience <- factor(c("Entry", "Mid", "Senior"),
                     levels = c("Entry", "Mid", "Senior"))
```

(c) Write a function called `prepend_level` that takes any vector of factor levels and prepends the text `"Level: "` to each level description. This function should be applicable to any factor levels, making it versatile for various data scenarios.

(d) (Bonus) What does the following code chunk do? Assume `prepend_level` is the same function that you devised in part (c) and `experience_reversed` is the object that you defined in part (b) above. (5 points)

```
forcats::fct_relabel(experience_reversed, prepend_level)
```