Homework 10 Solutions

Sarah Cannon

Due: 11/19/24

Question 1 (5 points)

> **Read the paper "Model Cards for Model Reporting," available at https://arxiv.org/pdf/1810.03993 (https://arxiv.org/pdf/1810.03993). In your own words in 1-2 paragraphs, you should:**

> **- explain what a model card is**

> **- explain why model cards are important**

> **- explain which element of a model card you were most surprised to see included and why it's important to include this in a model card.**

A model card is a document that accompanies machine learning models, detailing their intended use, performance across various conditions, and potential limitations. It includes metrics for different demographic groups, ethical considerations, and context-specific recommendations to ensure responsible deployment and informed usage.

Model cards are important because they promote transparency, fairness, and accountability in ML, particularly in sensitive applications like healthcare or law enforcement. They help users understand a model's suitability, mitigate biases, and ensure ethical practices in model development.

An element that stood out to me is "Intersectional Analysis" which evaluates performance across combined demographic factors like race and gender. This helps for identifying biases that may not appear when analyzing groups separately, ensuring fairness and equity in models.

Question 2 (3 points)

> **Load in the libraries necessary to make linear and spline models. Be sure to specify the same options we did in class, and explain why you should do this.**

```
library(modelr)
library(splines)
options(na.action = na.warn)
```

Question 3 (10 points)

The following code imports a data set with the daily confirmed cases of coronavirus in California, and makes a new column with the number of new daily cases, and focuses on March - July, 2020. For convenience, there is also a column with the day number, with March 1st beign 1, March 2nd being 2, etc.

```
c <- read_csv("https://raw.githubusercontent.com/datadesk/california-coronavirus-data/refs/heads/master/cdph-state-cases-deaths.csv") %>%
  select(date, confirmed_cases) %>%
  arrange(date) %>%
  mutate(new_daily_cases = confirmed_cases - lag(confirmed_cases)) %>%
  filter(date <= "2020-07-31", date >="2020-03-01") %>%
  mutate(day_number = row_number())
```

```
## Rows: 1215 Columns: 5
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl  (4): confirmed_cases, probable_cases, confirmed_and_probable_cases, confirmed_deaths
## date (1): date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
c
```

date <date>	confirmed_cases <dbl>	new_daily_cases <dbl>	day_number <int>
2020-03-01	567	98	1
2020-03-02	643	76	2
2020-03-03	728	85	3
2020-03-04	820	92	4
2020-03-05	931	111	5
2020-03-06	1090	159	6

date <date>	confirmed_cases <dbl>	new_daily_cases <dbl>	day_number <int>
2020-03-07	1254	164	7
2020-03-08	1433	179	8
2020-03-09	1763	330	9
2020-03-10	2121	358	10

1-10 of 153 rows Previous **1** 2 3 4 5 6 … 16 Next

a. (2 points) **Suppose the model $new_daily_cases = -100 + 50 * day_number$ is suggested. Add a column to your table from (a) that has the predicted number of new daily covid cases, based on this model.**

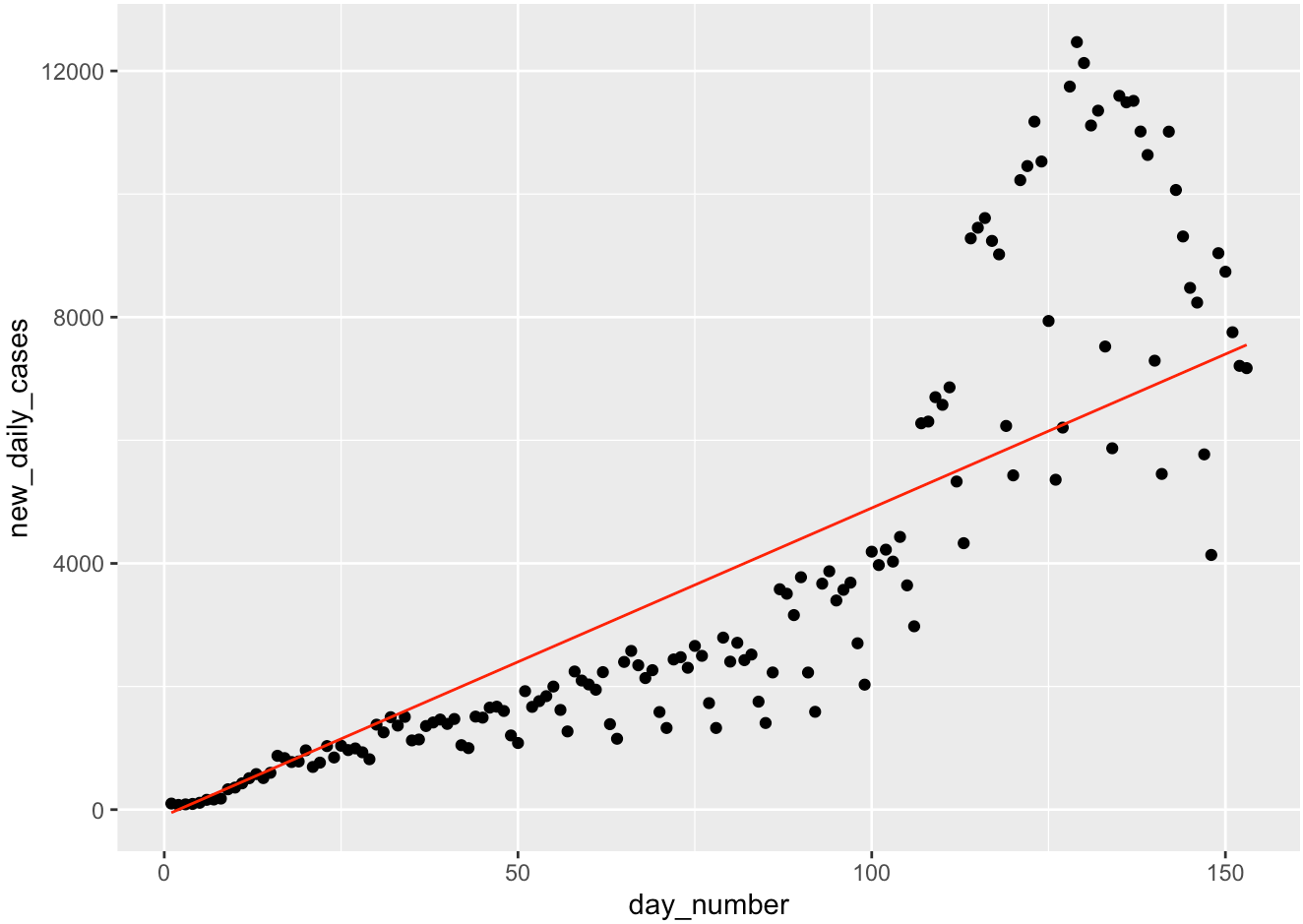
```
c <- c %>% mutate(predicted_new_daily_cases = -100 + 50 * day_number)
c
```

date <date>	confirmed_cases <dbl>	new_daily_cases <dbl>	day_number <int>	predicted_new_daily_cases <dbl>
2020-03-01	567	98	1	-50
2020-03-02	643	76	2	0
2020-03-03	728	85	3	50
2020-03-04	820	92	4	100
2020-03-05	931	111	5	150
2020-03-06	1090	159	6	200
2020-03-07	1254	164	7	250
2020-03-08	1433	179	8	300
2020-03-09	1763	330	9	350
2020-03-10	2121	358	10	400

1-10 of 153 rows Previous **1** 2 3 4 5 6 … 16 Next

b. (3 points) **Make a scatterplot with days_since_start on the x-axis and total number of new daily covid cases on the y-axis. Add a line to your plot for the model $new_daily_cases = -100 + 50 * day_number$. Does this seem like a good model?**

```
ggplot(c, aes(x = day_number, y = new_daily_cases)) +
  geom_point() +
  geom_line(aes(y = predicted_new_daily_cases), color = "red")
```



The model seems to be going in the right direction, but is linear while the data has non-linear trends.

c. (3 points) **Add to your table a column for the residuals of the model** $new_daily_cases = -100 + 50 * day_number$**.**

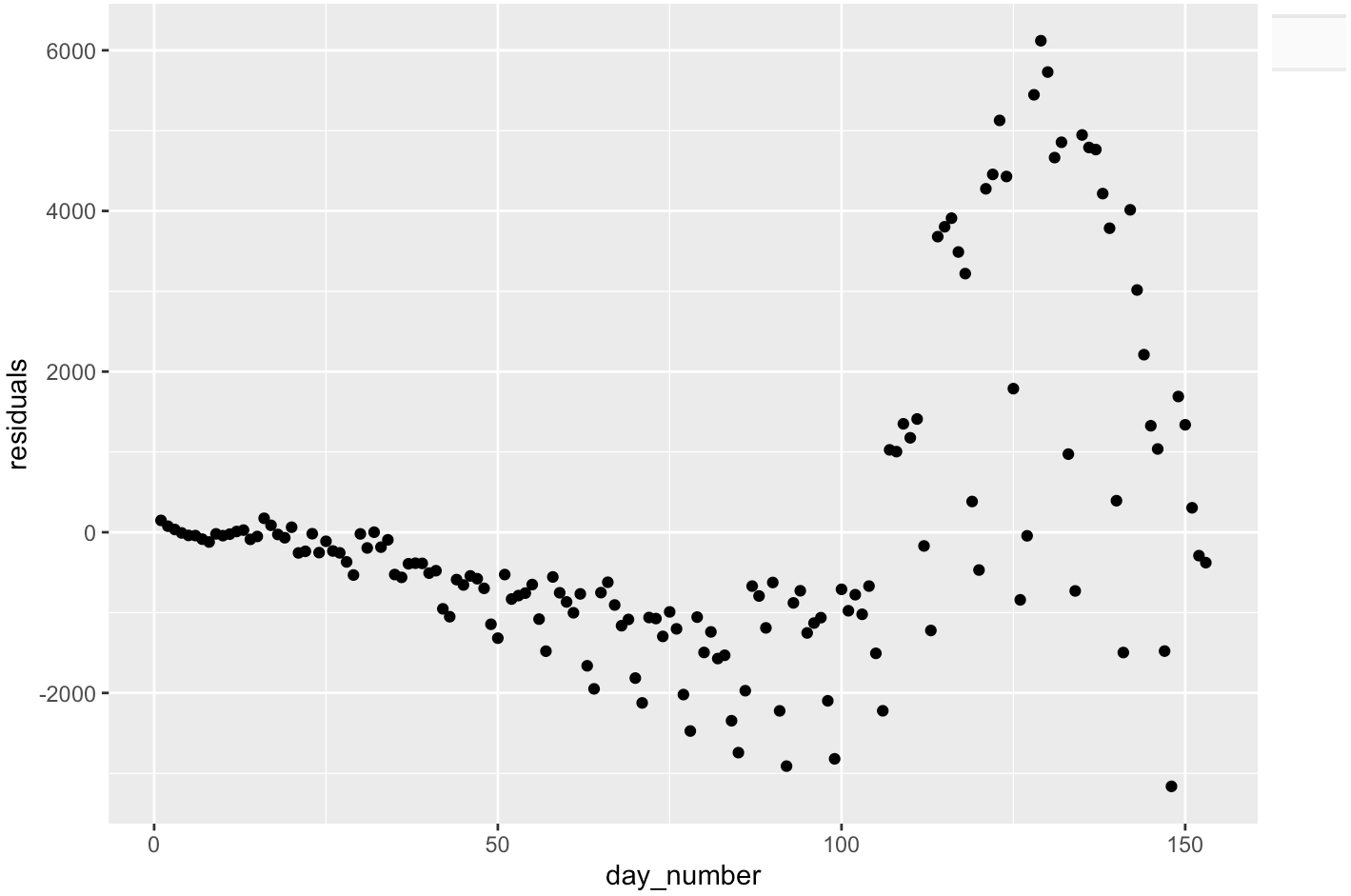
```
c <- c %>% mutate(residuals = new_daily_cases - predicted_new_daily_cases)
c
```

date <date>	confirmed_cases <dbl>	new_daily_cases <dbl>	day_num... <int>	predicted_new_daily_cases <dbl>	re
2020-03-01	567	98	1	-50	
2020-03-02	643	76	2	0	
2020-03-03	728	85	3	50	
2020-03-04	820	92	4	100	
2020-03-05	931	111	5	150	

date <date>	confirmed_cases <dbl>	new_daily_cases <dbl>	day_num… <int>	predicted_new_daily_cases <dbl>	re
2020-03-06	1090	159	6	200	
2020-03-07	1254	164	7	250	
2020-03-08	1433	179	8	300	
2020-03-09	1763	330	9	350	
2020-03-10	2121	358	10	400	

d. (2 points) **Make a scatterplot showing the days_since_start on the x-axis and the residuals on the y-axis. Are there any patterns in these residuals?**

```
ggplot(c, aes(x = day_number, y = residuals)) +
  geom_point()
```



The residuals first trend downwards, then significantly upwards, then significantly downwards.

Question 4 (18 points)

a. (2 points) **Using the same California covid data set from the previous question, use the lm function to come up with a different, better linear model for this data. Write out what the equation would be, e.g., in the form new_daily_cases = c1 + c2 * day_number, specifying what c1 and c2 are.**

```
model <- lm(new_daily_cases ~ day_number, data = c)
summary(model)
```

```
##
## Call:
## lm(formula = new_daily_cases ~ day_number, data = c)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4510.6 -1238.6  -156.3   936.8  5107.8
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1372.668    294.097  -4.667 6.68e-06 ***
## day_number     67.705      3.313  20.435  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1810 on 151 degrees of freedom
## Multiple R-squared:  0.7344, Adjusted R-squared:  0.7327
## F-statistic: 417.6 on 1 and 151 DF,  p-value: < 2.2e-16
```

new_daily_cases = -1372.668 + 67.705 * day_number

b. (2 points) **Using functions we learned in class rather than doing the calculations explicitly, add a column for the predictions of your model from (a) and a column for the residuals of your model from (a) on to your data set.**

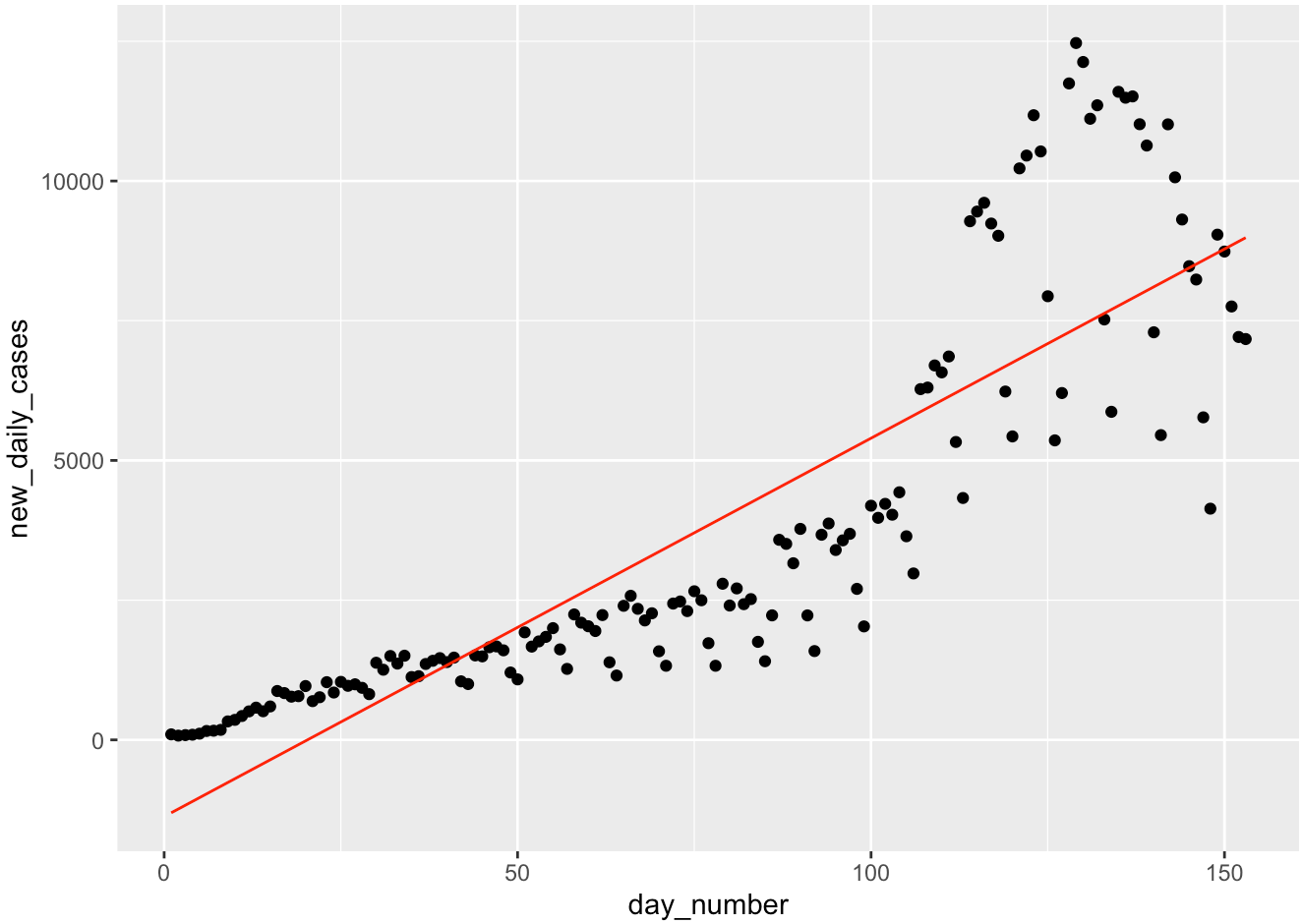
```
c <- c %>% mutate(predicted_new_daily_cases = -1372.668 + 67.705 * day_number,
                  residuals = new_daily_cases - predicted_new_daily_cases)
c
```

date	confirmed_cases	new_daily_cases	day_num...	predicted_new_daily_cases	re
<date>	<dbl>	<dbl>	<int>	<dbl>	
2020-03-01	567	98	1	-1304.963	1

date <date>	confirmed_cases <dbl>	new_daily_cases <dbl>	day_num... <int>	predicted_new_daily_cases <dbl>	re
2020-03-02	643	76	2	-1237.258	1
2020-03-03	728	85	3	-1169.553	1
2020-03-04	820	92	4	-1101.848	1
2020-03-05	931	111	5	-1034.143	1
2020-03-06	1090	159	6	-966.438	1
2020-03-07	1254	164	7	-898.733	1
2020-03-08	1433	179	8	-831.028	1
2020-03-09	1763	330	9	-763.323	1
2020-03-10	2121	358	10	-695.618	1

c. (3 points) **Make a plot showing the data and your model from (a), and then make a plot showing the residuals of this model. Assess this model: how good is it?**

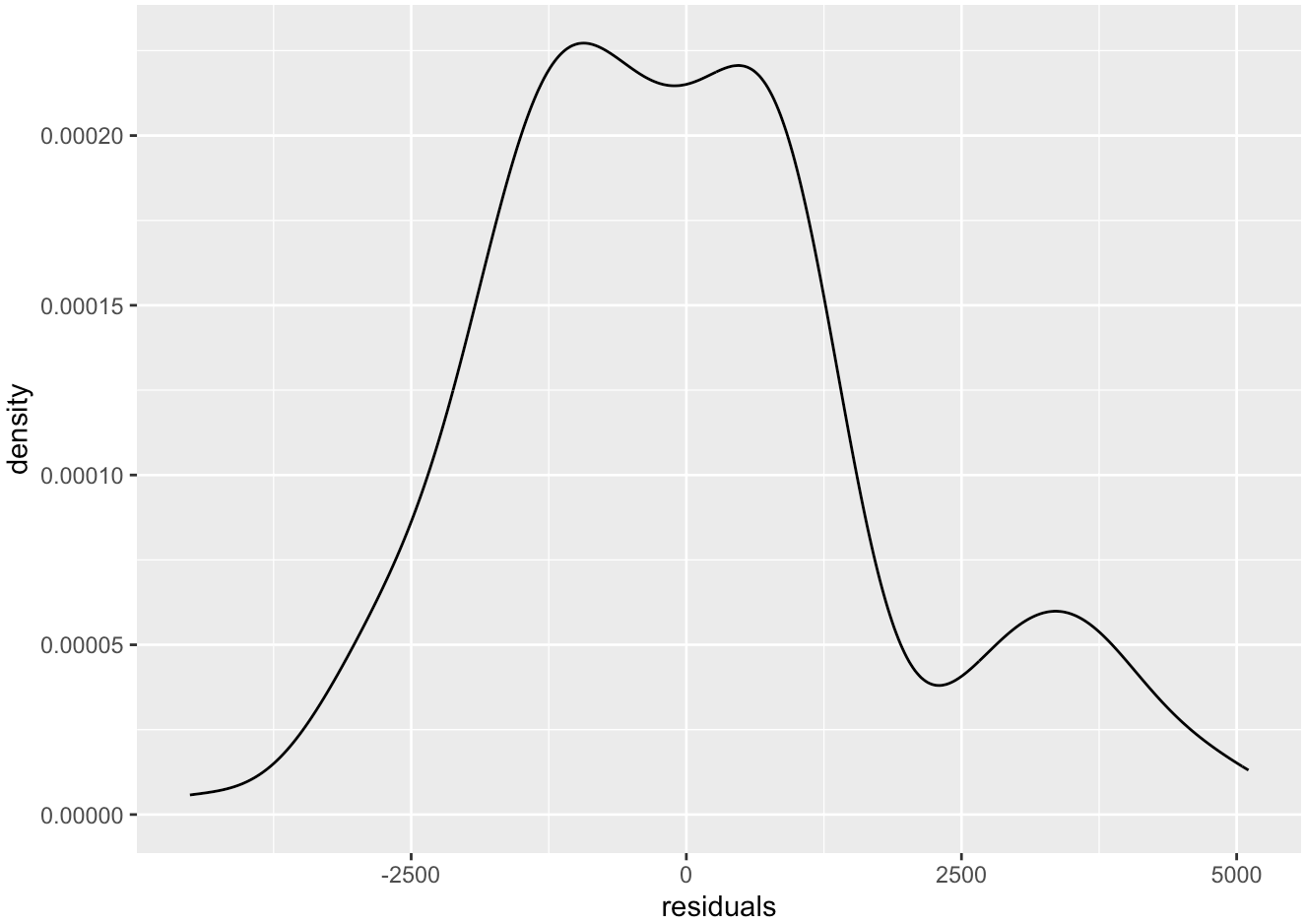
```
ggplot(c, aes(x = day_number, y = new_daily_cases)) +
  geom_point() +
  geom_line(aes(y = predicted_new_daily_cases), color = "red")
```



The model seems to fit better, but is still limited by the linear nature of the model.

> d. (3 points) **Make a density plot of the residuals. Explain what this plot should look like if your model is a good model, and compare the plot you've made to this ideal.**

```
ggplot(c, aes(x = residuals)) +
  geom_density()
```



The plot should be centered around 0 with a symmetric distribution. The actual plot is centered around 0, but is not symmetric.

e. (3 points) **What is the R^2 value for this model, and what is the correlation for this model? Explain what this tells you about the model, and about how much of the variance in new_daily_cases can be predicted using day_number.**

R^2 = 0.7344 Correlation = 0.857

This tells us that 73.44% of the variance in new_daily_cases can be predicted using day_number.

f. (3 points) **While we should always be careful about the limitations of our model, what does your linear model predict to be the total number of new daily covid cases on August 3 (day number 156)? See how close your model was to the reality (for example, by changing the filter() on the chunk importing the code; just be sure to change it back after you're done!); was this a good prediction?**

```
predicted_cases <- -1372.668 + 67.705 * 156
predicted_cases
```

```
## [1] 9189.312
```

Actual cases on August 3, 2020:

```
c <- read_csv("https://raw.githubusercontent.com/datadesk/california-coronavirus-data/refs/heads/master/cdph-state-cases-deaths.csv") %>%
  select(date, confirmed_cases) %>%
  arrange(date) %>%
  mutate(new_daily_cases = confirmed_cases - lag(confirmed_cases)) %>%
  filter(date <= "2020-08-03", date >= "2020-03-01") %>%
  mutate(day_number = row_number())
```

```
## Rows: 1215 Columns: 5
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl  (4): confirmed_cases, probable_cases, confirmed_and_probable_cases, confirmed_deaths
## date (1): date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
actual_cases_august_3 <- c %>% filter(date == "2020-08-03") %>% select(new_daily_cases)
actual_cases_august_3
```

new_daily_cases
<dbl>
7527

1 row

Difference between predicted and actual cases:

```
predicted_cases - actual_cases_august_3
```

new_daily_cases
<dbl>
1662.312

1 row

The model was off by 1662 cases.

g. (2 points) **Explain why using your model to predict the number of covid cases on August 3, 2020 might be a reasonable thing to do, but using your model to predict the number of daily covid cases a year later, on August 3, 2021 (365 + 156 = 521 days after the conflict started) would not be a reasonable thing to do.**

Using the model to predict cases on August 3, 2020, is reasonable because it falls nearby the data range used to build the model, capturing local trends.

Predicting for August 3, 2021, is unreasonable because extending predictions far beyond the data range can lead to inaccuracies. Factors like new variants and vaccination rates can alter trends, which the model doesn't account for.

Question 5 (12 points)

a. (3 points) **The lm function finds the linear model such that the sum of the squares of the residuals is as small as possible. Compute the sum of the squares of the residuals of both the model -100 + 50 * day_number from Question 3 and the model you found using the lm() function in Question 4. Is the sum of the squares of the residuals smaller for the model you found with the lm function?**

```
model1_residuals <- c$new_daily_cases - (-100 + 50 * c$day_number)
sum_squares_model1 <- sum(model1_residuals^2)
sum_squares_model1
```

```
## [1] 611366599
```

[1] 611366599

Q 4 model:

```
model2_residuals <- c$new_daily_cases - predict(model, c)
sum_squares_model2 <- sum(model2_residuals^2)
sum_squares_model2
```

```
## [1] 542179146
```

[1] 542179146

The sum of the squares of the residuals is smaller for the model found using the lm() function, indicating that it fits the data better.

b. (3 points) **The sum of the square of the residuals is not the only way to assess a model. For the same two models, compute the sum of the absolute values of the residuals. According to this metric, which model is better?**

```
sum_abs_model1 <- sum(abs(model1_residuals))
sum_abs_model1
```

```
## [1] 214191
```

[1] 214191

```
sum_abs_model2 <- sum(abs(model2_residuals))
sum_abs_model2
```

```
## [1] 226128.5
```

[1] 226128.5

According to the sum of absolute values of the residuals, Model 1 is better, as it has a smaller sum. This suggests that Model 1 might be less sensitive to outliers or has a more consistent fit across the data points.

c. (3 points) **Based on everything you've considered so far about the two models (plotting the models, plotting the residuals, the values computed in parts a and b), which model do you think is better? Explain why.**

The choice of the better model depends on the context and what is prioritize. If minimizing large errors is important, Model 2 is better. If robustness to outliers is more important, Model 1 might be better. Model 2 is typically preferred.

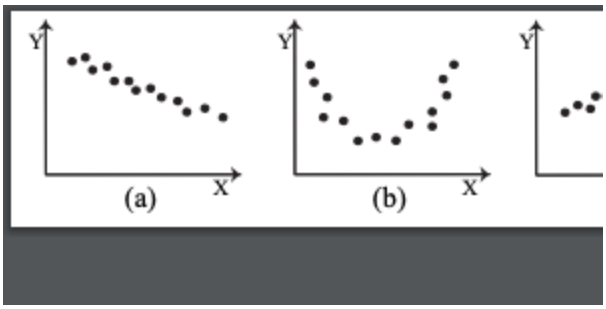
d. (3 points) **Do some research yourself on the difference between modelling to try to minimize the sum of the squares of the residuals vs. the sum of the absolute values of the residuals. When might you want to use one vs. the other? Explain in your own words here.**

Sum of Squares of Residuals is sensitive to outliers and is commonly used because it has nice mathematical properties, making it easier to compute derivatives and find the best fit.

Sum of Absolute Values of Residuals is more robust to outliers and might be preferred when the data contains outliers or when a more robust fit is desired.

Question 6 (3 points)

Consider the five scatterplots of X and Y shown in the correlation.pdf document, attached to this assignment (note: you'll need to put the correlation.pdf file in the same directory as this RMarkdown file to make this PDF show up in your knitted file)



Correlation Image

Put these plots in order from smallest correlation (closest to -1) to largest correlation (closest to +1), by listing the letters a-e in that order here:

a, d, b, e, c

Question 7 (12 points)

Consider the data daily_covid_cases_winter_peak, made by the following code chunk. This shows the number of new daily (confirmed and probable) covid cases in LA County from October 15, 2022 to February 1, 2023.

```
# You do not need to know how this code works
daily_covid_cases_winter_peak <- read_csv("https://raw.githubusercontent.com/datadesk/ca
lifornia-coronavirus-data/master/cdph-county-cases-deaths.csv") %>%
  filter(county == "Los Angeles") %>%
  arrange(date) %>%
  mutate(new_daily_cases =  confirmed_and_probable_cases - lag(confirmed_and_probable_ca
ses, 1)) %>%
  filter(date >= "2022-10-15", date <= "2023-02-01" ) %>%
  mutate(day = row_number()) %>%
  select(day, date, new_daily_cases)
```

```
## Rows: 70470 Columns: 8
## ── Column specification ────────────────────────────────────────────────────
──────────────────────
## Delimiter: ","
## chr  (2): county, fips
## dbl  (5): population, confirmed_cases, probable_cases, confirmed_and_probable_cases,
confirmed_deaths
## date (1): date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
daily_covid_cases_winter_peak
```

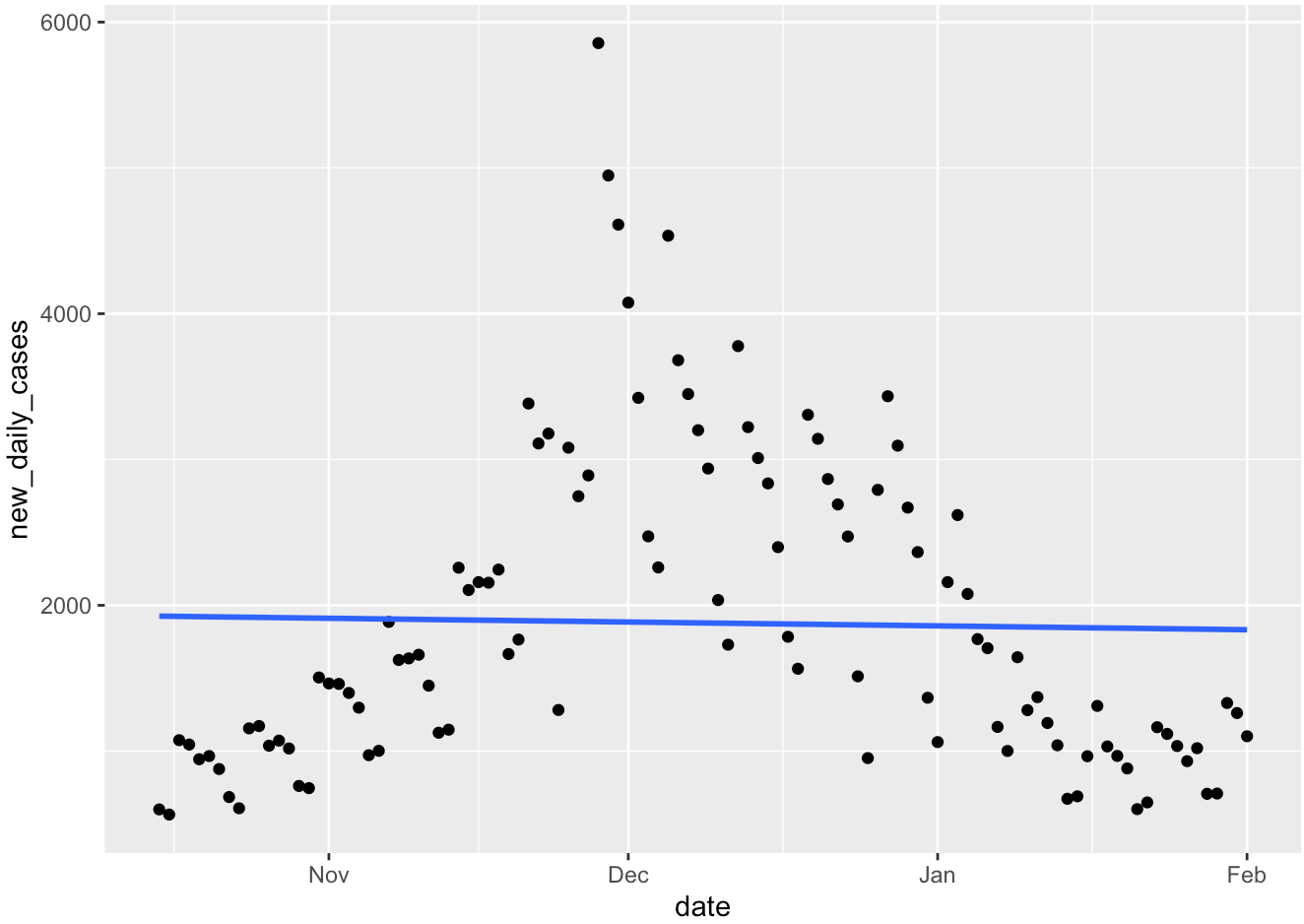
day <int>	date <date>	new_daily_cases <dbl>
1	2022-10-15	600
2	2022-10-16	565
3	2022-10-17	1075
4	2022-10-18	1045
5	2022-10-19	944
6	2022-10-20	966
7	2022-10-21	877
8	2022-10-22	685
9	2022-10-23	608
10	2022-10-24	1156

1-10 of 110 rows Previous **1** 2 3 4 5 6 … 11 Next

a. (3 points) **Make a scatterplot of the relationship between date (on the x-axis) and new_daily_cases (on the y-axis). Make a linear model for this data and plot its residuals. Explain why a linear model is not a good choice for this data.**

```
ggplot(daily_covid_cases_winter_peak, aes(x = date, y = new_daily_cases)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



A linear model is not a good choice for this data because the relationship between date and new_daily_cases is not linear.

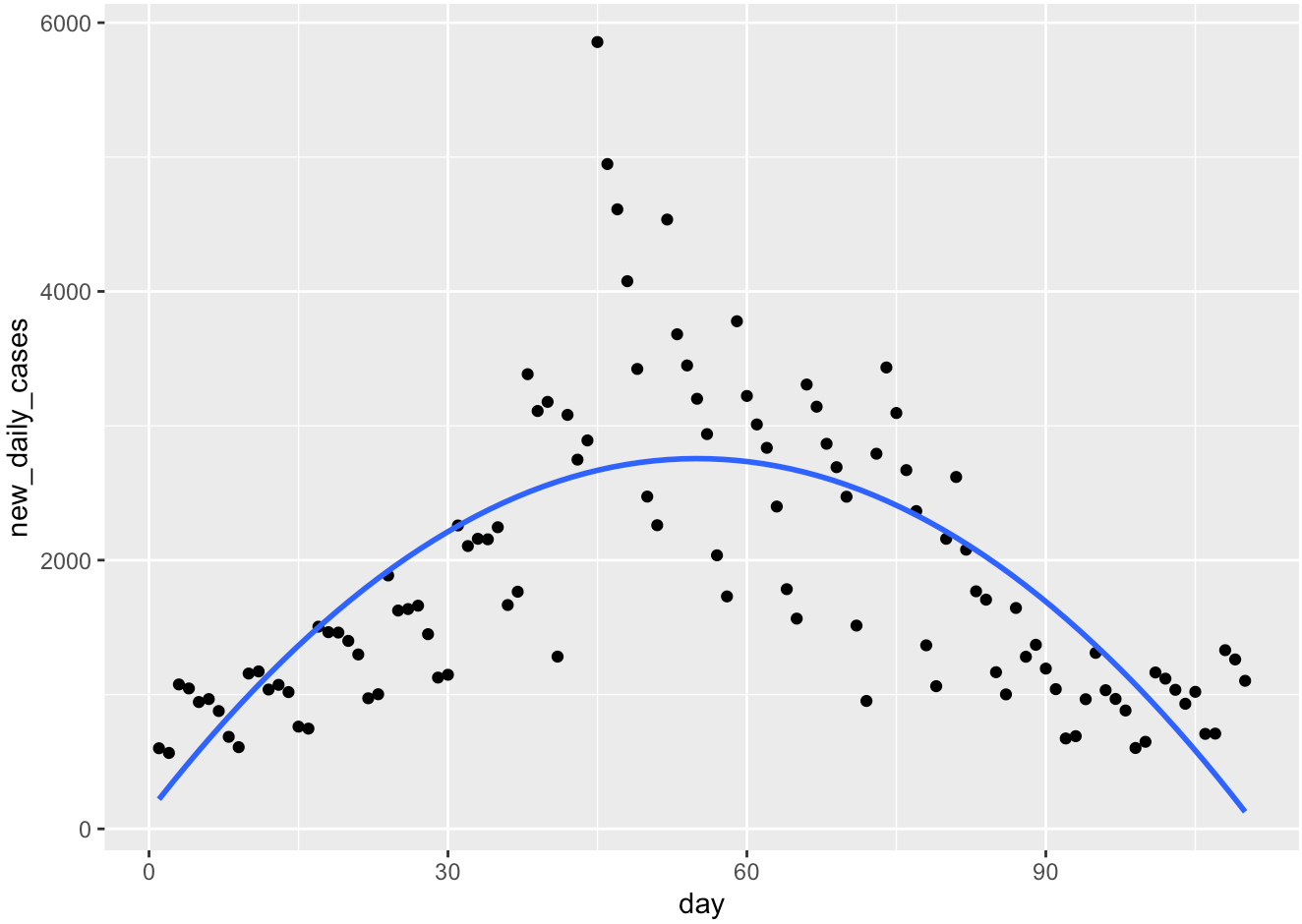
b. (4 points) **Create a quadratic model for this data, trying to predict new_daily_cases using day (be sure to use day instead of date, because the lm function will not work with a data type). Write out the equation of your model, draw your model on top of your data, and make a scatterplot of its residuals. Assess the strengths and weaknesses of this model, referring to your plots.**

```
quadratic_model <- lm(new_daily_cases ~ day + I(day^2), data = daily_covid_cases_winter_
peak)
summary(quadratic_model)
```

```
##
## Call:
## lm(formula = new_daily_cases ~ day + I(day^2), data = daily_covid_cases_winter_peak)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1552.7  -464.9   -71.9   351.1  3187.4
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 125.36146  218.84095   0.573    0.568
## day          95.63349    9.10108  10.508   <2e-16 ***
## I(day^2)     -0.86926    0.07943 -10.943   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 751.2 on 107 degrees of freedom
## Multiple R-squared:  0.5284, Adjusted R-squared:  0.5196
## F-statistic: 59.95 on 2 and 107 DF,  p-value: < 2.2e-16
```

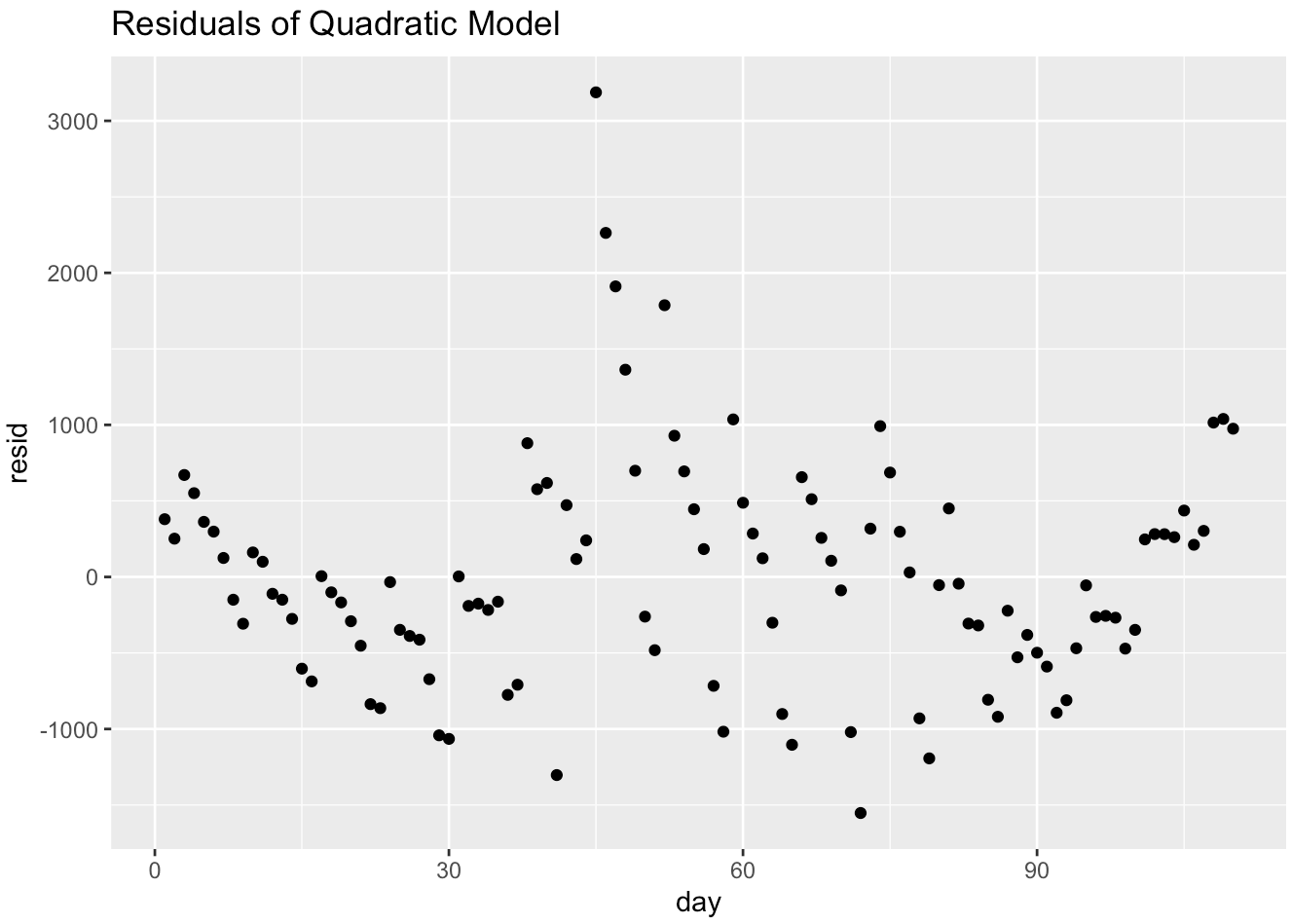
Model Equation: new_daily_cases = 125.3 + 95.6 * day + (-0.87 * day^2)

```
ggplot(daily_covid_cases_winter_peak, aes(x = day, y = new_daily_cases)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), se = FALSE)
```



Residuals:

```
daily_covid_cases_winter_peak <- daily_covid_cases_winter_peak %>% add_residuals(quadrat
ic_model)

ggplot(daily_covid_cases_winter_peak, aes(x = day, y = resid)) +
  geom_point() +
  labs(title = "Residuals of Quadratic Model")
```



The quadratic model is a better fit than the linear model because it captures the non-linear trend in the data. Although it is still not perfect, it is a better fit than the linear model.

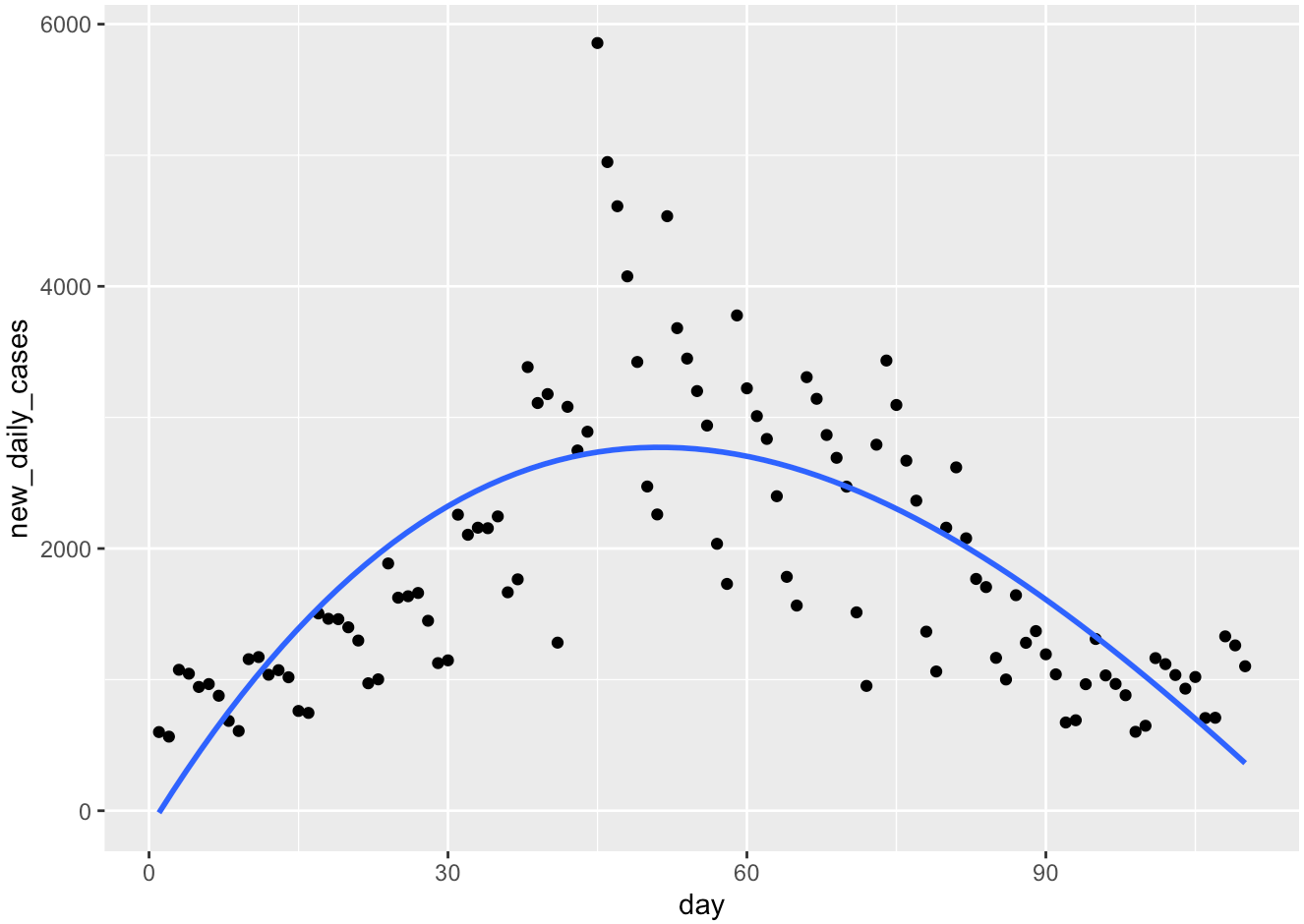
c. (5 points) **Look at the residuals both for your linear and quadratic model. Based on these residuals, what polynomial model do you think might be appropriate? Explain your choice, make your model, draw it on top of your data, and plot your residuals. Assess this model - is it a good model? Is it better than your linear and quadratic models?**

Since the trend of the residuals starts out near zero with slight negative slope, then flat, then rapidly increasing, then rapidly decreasing, then flat again, then slightly positive slope, a cubic model might be appropriate.

```
cubic_model <- lm(new_daily_cases ~ day + I(day^2) + I(day^3), data = daily_covid_cases_
winter_peak)
summary(cubic_model)
```

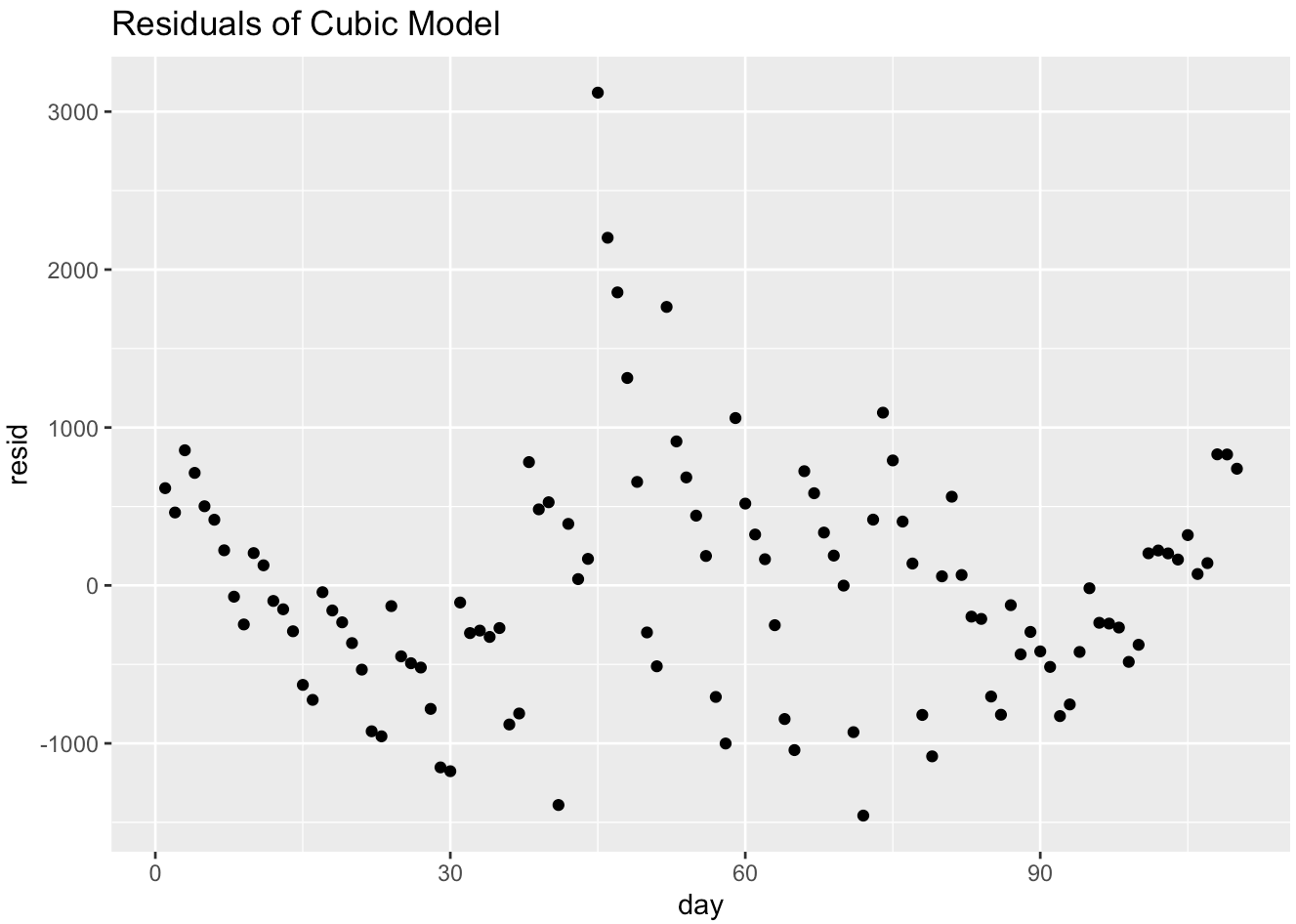
```
##
## Call:
## lm(formula = new_daily_cases ~ day + I(day^2) + I(day^3), data = daily_covid_cases_wi
nter_peak)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1457.34  -475.31   -84.66   416.48  3120.32
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.379e+02  2.955e+02  -0.467  0.64181
## day          1.235e+02  2.295e+01   5.379 4.51e-07 ***
## I(day^2)    -1.493e+00  4.793e-01  -3.115  0.00236 **
## I(day^3)     3.748e-03  2.839e-03   1.320  0.18972
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 748.6 on 106 degrees of freedom
## Multiple R-squared:  0.536,  Adjusted R-squared:  0.5229
## F-statistic: 40.82 on 3 and 106 DF,  p-value: < 2.2e-16
```

```
ggplot(daily_covid_cases_winter_peak, aes(x = day, y = new_daily_cases)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2) + I(x^3), se = FALSE)
```



Residuals:

```
daily_covid_cases_winter_peak <- daily_covid_cases_winter_peak %>% add_residuals(cubic_m
odel)

ggplot(daily_covid_cases_winter_peak, aes(x = day, y = resid)) +
  geom_point() +
  labs(title = "Residuals of Cubic Model")
```



I would say the cubic model is not significantly better than the quadratic model, although it's residuals are slightly more centered around zero and it's R^2 is slightly higher.

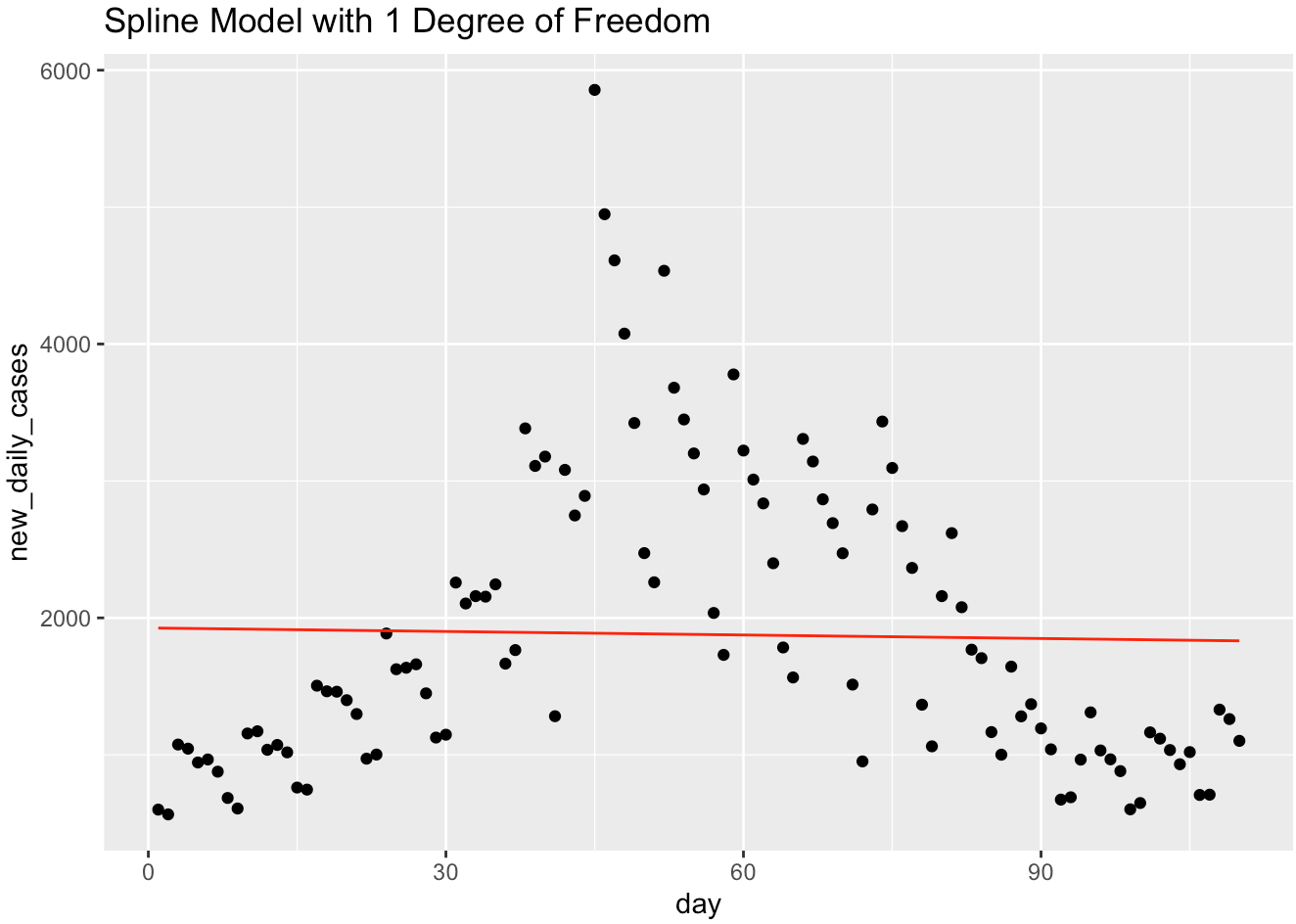
Question 8 (15 points)

This question continues looking at the data set daily_covid_cases_winter_peak from the previous question.

a. (2 points) **Create a model for this data that is a spline with one degree of freedom. Make predictions for your model, and draw your model on top of the data points. What do you observe?**

```
spline_model_1 <- lm(new_daily_cases ~ ns(day, df = 1), data = daily_covid_cases_winter_
peak)
```

```
daily_covid_cases_winter_peak <- daily_covid_cases_winter_peak %>% add_predictions(splin
e_model_1)

ggplot(daily_covid_cases_winter_peak, aes(x = day, y = new_daily_cases)) +
  geom_point() +
  geom_line(aes(y = pred), color = "red") +
  labs(title = "Spline Model with 1 Degree of Freedom")
```

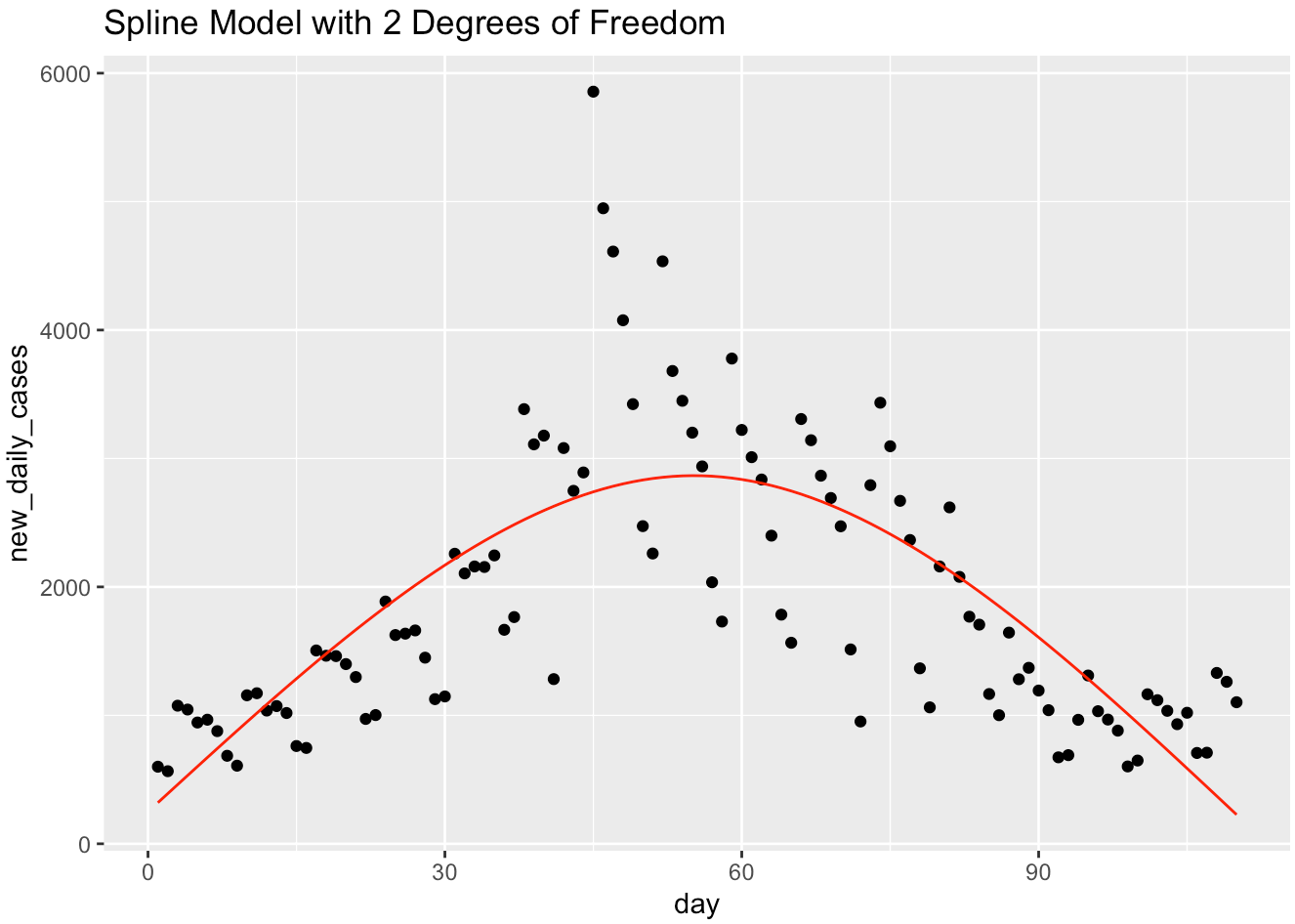


With one degree of freedom, the spline is essentially a linear model. It may not capture the data non-linear trends well.

b. (2 points) **Create a model for this data that is a spline with two degrees of freedom. Make predictions for your model, and draw your model on top of the data points. What do you observe?**

```
spline_model_2 <- lm(new_daily_cases ~ ns(day, df = 2), data = daily_covid_cases_winter_
peak)
```

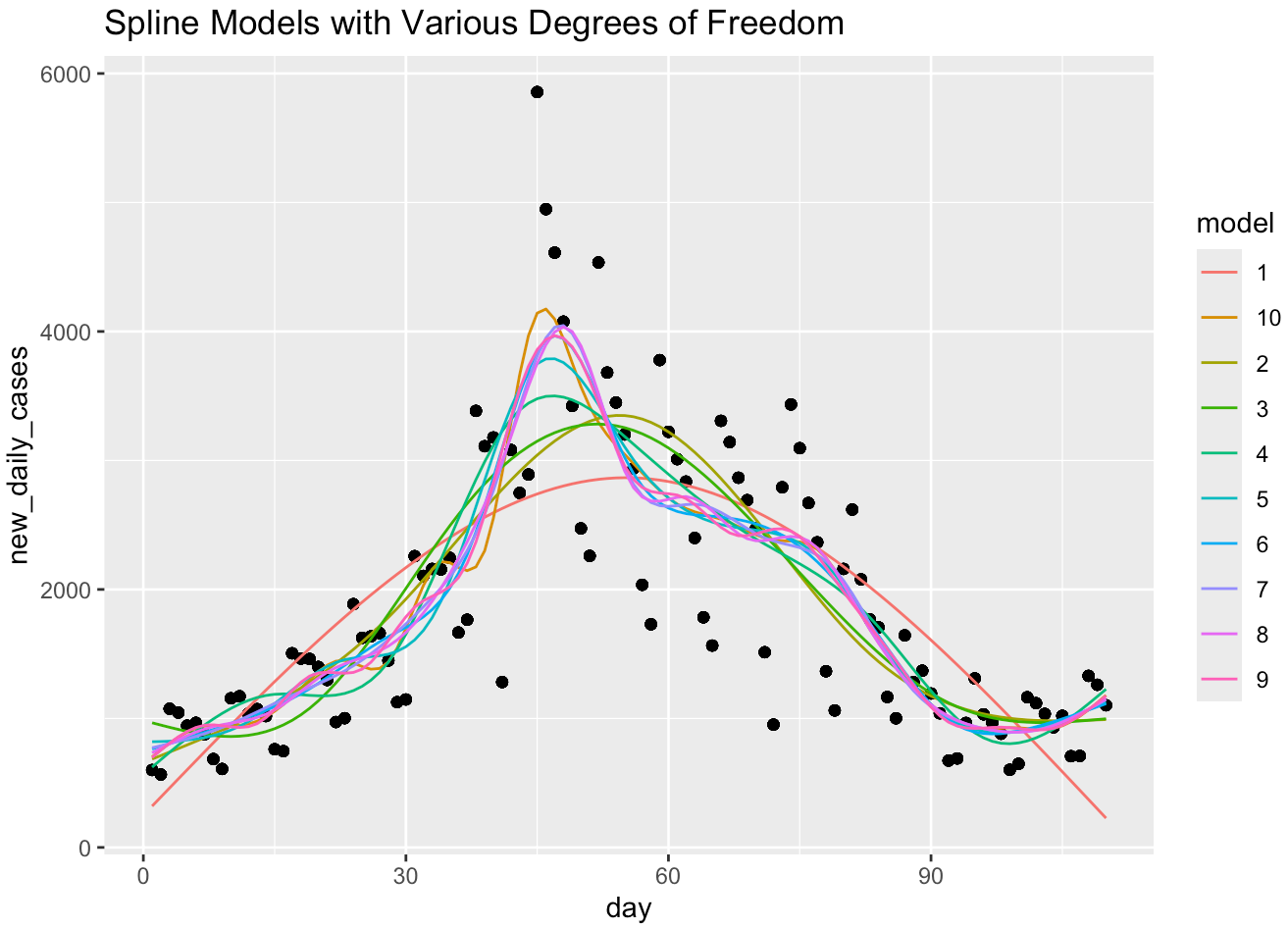
```
daily_covid_cases_winter_peak <- daily_covid_cases_winter_peak %>% add_predictions(splin
e_model_2)

ggplot(daily_covid_cases_winter_peak, aes(x = day, y = new_daily_cases)) +
  geom_point() +
  geom_line(aes(y = pred), color = "red") +
  labs(title = "Spline Model with 2 Degrees of Freedom")
```



With two degrees of freedom, the spline can capture some curvature, providing a better fit than a linear model.

c. (4 points) **Create 10 models for this data for splines with 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 degrees of freedom. Use one single function to add predictions for *all 10* of these data points onto your original data set, and plot all 10 of these models. Hint: You can do this using one call to the ggplot() function.**

```
models <- map(seq(2, 20, by = 2), ~ lm(new_daily_cases ~ ns(day, df = .x), data = daily_
covid_cases_winter_peak))

predictions <- map_df(models, ~ add_predictions(daily_covid_cases_winter_peak, .x), .id
= "model")

ggplot(predictions, aes(x = day, y = new_daily_cases)) +
  geom_point() +
  geom_line(aes(y = pred, color = model)) +
  labs(title = "Spline Models with Various Degrees of Freedom")
```



d. (2 points) **Based on your plots in the previous part, which of the 10 models would you choose if you had to pick one?**

I might use model 5, as it seems to capture the trend well without too much overfitting, although this visual decision is somewhat subjective.

e. (2 points) **Explain why you don”t just want to use as many degrees of freedom as possible, for example why using a spline with 100 degrees of freedom for this data would be a bad idea.**

Using too many degrees of freedom can lead to overfitting, where the model captures noise rather than the underlying trend.

f. (3 points) **The following data set is all new daily covid cases in California from February 1, 2020 to May 30, 2023 (when data collection stopped). Create an appropriate model for this data, draw it on top of the data, and explain why you chose the model you did.**

```
c2 <- read_csv("https://raw.githubusercontent.com/datadesk/california-coronavirus-data/refs/heads/master/cdph-state-cases-deaths.csv") %>%
  select(date, confirmed_cases) %>%
  arrange(date) %>%
  mutate(new_daily_cases = confirmed_cases - lag(confirmed_cases)) %>%
  mutate(day_number = row_number())
```

```
## Rows: 1215 Columns: 5
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl  (4): confirmed_cases, probable_cases, confirmed_and_probable_cases, confirmed_deaths
## date (1): date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
c2
```

date <date>	confirmed_cases <dbl>	new_daily_cases <dbl>	day_number <int>
2020-02-01	25	*NA*	1
2020-02-02	34	9	2
2020-02-03	41	7	3
2020-02-04	44	3	4
2020-02-05	47	3	5
2020-02-06	53	6	6
2020-02-07	84	31	7
2020-02-08	91	7	8
2020-02-09	96	5	9
2020-02-10	102	6	10

1-10 of 1,215 rows Previous **1** 2 3 4 5 6 … 122 Next

```
spline_model <- lm(new_daily_cases ~ ns(day_number, df = 10), data = c2)
```

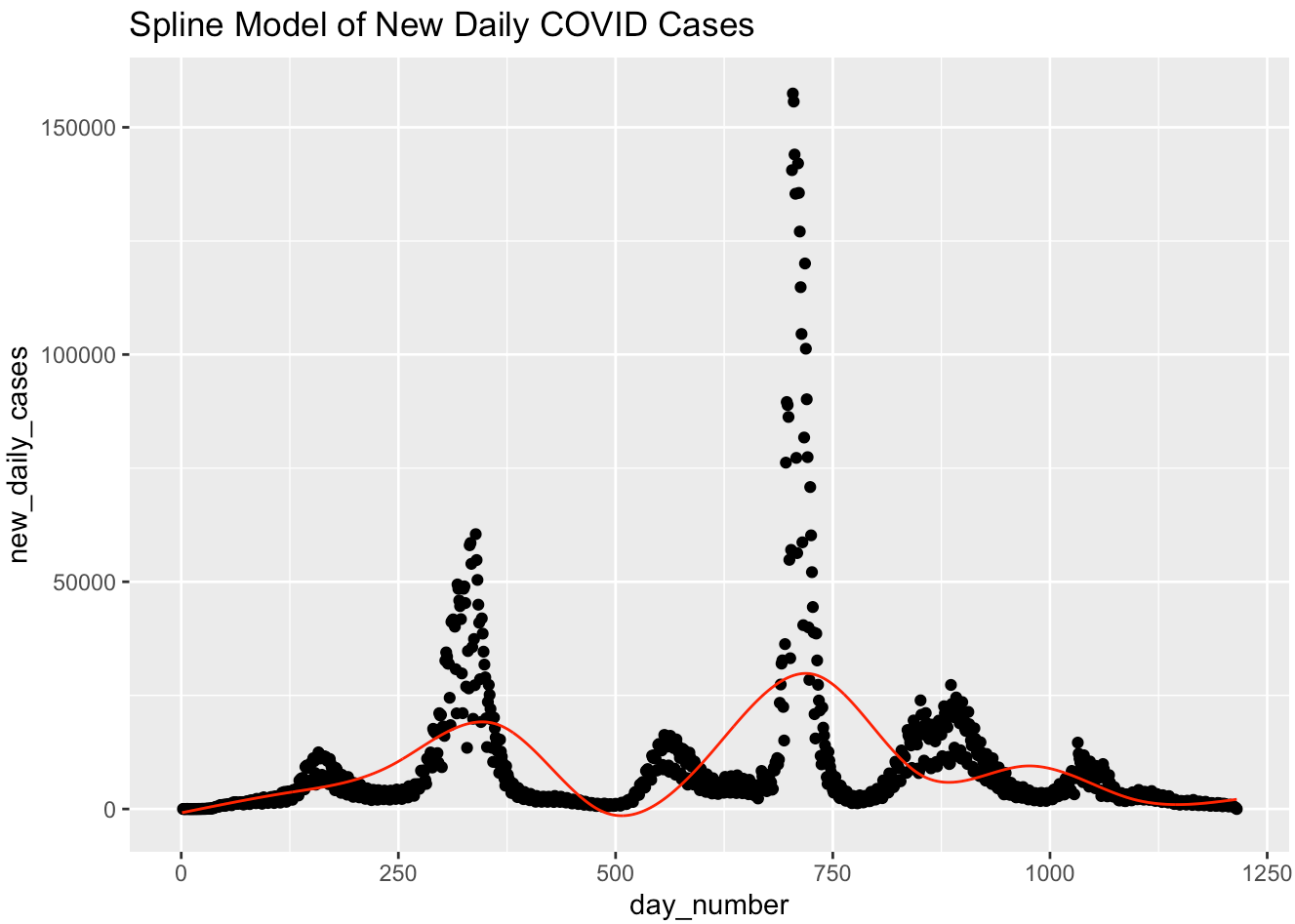
```
## Warning: Dropping 1 rows with missing values
```

```
c2 <- c2 %>% add_predictions(spline_model)
```

Plot:

```
ggplot(c2, aes(x = day_number, y = new_daily_cases)) +
  geom_point() +
  geom_line(aes(y = pred), color = "red") +
  labs(title = "Spline Model of New Daily COVID Cases")
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range (`
geom_point()`).
```



A spline model with 10 degrees of freedom is chosen to balance capturing the datas trends without overfitting. The plot shows how well the model fits the data, highlighting trends over time.

Question 9 (16 points)

This question looks at the effect of weekday on the number of new reported confirmed covid cases. This code creates a new dataset cw which has a new column for the weekday of each particular date.

```
cw <- c %>% mutate(weekday = factor(weekdays(date),
                    levels = c("Monday", "Tuesday", "Wednesday", "Thursday",
                    "Friday", "Saturday", "Sunday") ))
cw
```

date <date>	confirmed_cases <dbl>	new_daily_cases <dbl>	day_number <int>	weekday <fct>
2020-03-01	567	98	1	Sunday
2020-03-02	643	76	2	Monday
2020-03-03	728	85	3	Tuesday
2020-03-04	820	92	4	Wednesday
2020-03-05	931	111	5	Thursday
2020-03-06	1090	159	6	Friday
2020-03-07	1254	164	7	Saturday
2020-03-08	1433	179	8	Sunday
2020-03-09	1763	330	9	Monday
2020-03-10	2121	358	10	Tuesday

1-10 of 156 rows Previous **1** 2 3 4 5 6 … 16 Next

a. (2 points) **Use the lm function to create a linear model that predicts the number of new daily cases for each weekday. You model should *only* use weekday to make this prediction and not days since start. Output this linear model.**

```
linear_model <- lm(new_daily_cases ~ weekday, data = cw)
summary(linear_model)
```

```
##
## Call:
## lm(formula = new_daily_cases ~ weekday, data = cw)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4496  -2474  -1243   2714   8018
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4572.0      719.9   6.351 2.45e-09 ***
## weekdayTuesday     -121.3     1029.6  -0.118   0.9064
## weekdayWednesday   -143.4     1029.6  -0.139   0.8894
## weekdayThursday    -356.9     1029.6  -0.347   0.7294
## weekdayFriday      -394.9     1029.6  -0.384   0.7019
## weekdaySaturday   -1609.3     1029.6  -1.563   0.1202
## weekdaySunday     -2235.8     1018.1  -2.196   0.0296 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3452 on 149 degrees of freedom
## Multiple R-squared:  0.05458,    Adjusted R-squared:  0.01651
## F-statistic: 1.434 on 6 and 149 DF,  p-value: 0.2054
```

b. (3 points) **How many coefficients does your model have and why? What are the variables that correspond to these coefficients? Explain what each of these variables means and what values they can take on.**

The model has 7 coefficients: one for each weekday (Monday is the baseline) and the intercept. The intercept is the coefficient for Monday, while the other coefficients are the differences between each weekday and Monday.

c. (3 points) **Write out the equation for your model, and use this equation (not code) to predict the number of new daily cases on Friday and the number of new daily cases on Monday.**

new_daily_cases = 4572.0 + (-394.9) * Friday + 0 * Monday

Predicted number of new daily cases on Friday: new_daily_cases = 4572.0 + (-394.9) * 1 = 4177.1

Predicted number of new daily cases on Monday: new_daily_cases = 4572.0 + 0 * 1 = 4572.0

d. (3 points) **Create a data grid that has a column for each weekday and add a column for the predicted number of new daily cases for each weekday. (Hint: go back and make sure your answers to (c) match what you see here!)**

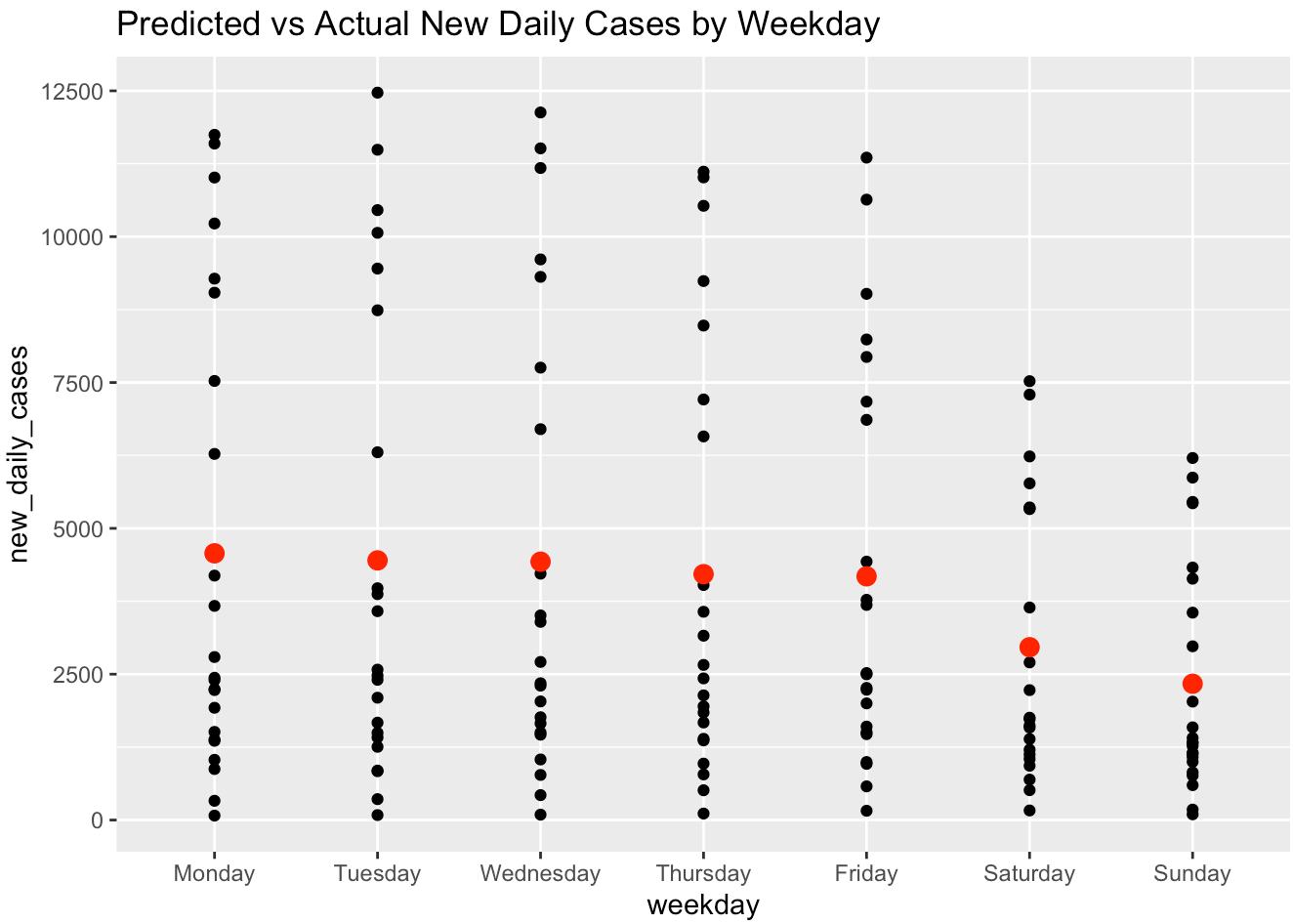
```
data_grid <- data.frame(weekday = levels(cw$weekday))

data_grid$predicted_cases <- predict(linear_model, newdata = data_grid)
data_grid
```

weekday <chr>	**predicted_cases** <dbl>
Monday	4571.957
Tuesday	4450.682
Wednesday	4428.545
Thursday	4215.091
Friday	4177.091
Saturday	2962.682
Sunday	2336.174

7 rows

e. (3 points) **Plot both the original data points and your predictions with an appropriate type of plot.**

```
ggplot(cw, aes(x = weekday, y = new_daily_cases)) +
  geom_point() +
  geom_point(data = data_grid, aes(y = predicted_cases), color = "red", size = 3) +
  labs(title = "Predicted vs Actual New Daily Cases by Weekday")
```



f. (2 points) **Compute the average daily number of new cases for each weekday. What do you notice?**

```
average_cases <- cw %>%
  group_by(weekday) %>%
  summarize(average_cases = mean(new_daily_cases, na.rm = TRUE))
average_cases
```

weekday <fct>	**average_cases** <dbl>
Monday	4571.957
Tuesday	4450.682
Wednesday	4428.545
Thursday	4215.091
Friday	4177.091
Saturday	2962.682
Sunday	2336.174

7 rows

The average number of new daily cases decreases from Monday to Sunday. There is a noticeable drop in cases over the weekend, which might indicate reporting delays or lower testing rates on weekends.

Question 10 (6 points)

a. (3 points) **Explain why some variables could be categorical or numerical depending on the context, and give an example of this different from the examples we discussed in class (an the example in part b).**

variables can switch between being categorical or numerical depending on the situation. Take “age” for instance: it can be a category like “teen” or “adult” in some cases, or just a number like 25 or 30 in others. It all depends on what youre looking at.

b. (3 points) **The following data includes how many colors Bob Ross used in each episode of “The Joy of Painting.”**

```
br <- read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2023/2023-02-21/bob_ross.csv')
```

```
## Rows: 403 Columns: 27
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): img_src, painting_title, youtube_src, colors, color_hex
## dbl  (4): painting_index, season, episode, num_colors
## lgl (18): Black_Gesso, Bright_Red, Burnt_Umber, Cadmium_Yellow, Dark_Sienna, Indian_Red, Indian_Yello...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
br %>% select(season, episode, num_colors)
```

season <dbl>	episode <dbl>	num_colors <dbl>
1	1	8
1	2	8
1	3	9

season <dbl>	episode <dbl>	num_colors <dbl>
1	4	3
1	5	8
1	6	4
1	7	8
1	8	8
1	9	8
1	10	8

1-10 of 403 rows Previous **1** 2 3 4 5 6 … 41 Next

```
br
```

painting_index <dbl>	img_src <chr>	painting_title <chr>
282	https://www.twoinchbrush.com/images/painting282.png	A Walk in the Woo
283	https://www.twoinchbrush.com/images/painting283.png	Mt. McKinley
284	https://www.twoinchbrush.com/images/painting284.png	Ebony Sunset
285	https://www.twoinchbrush.com/images/painting285.png	Winter Mist
286	https://www.twoinchbrush.com/images/painting286.png	Quiet Stream
287	https://www.twoinchbrush.com/images/painting287.png	Winter Moon
288	https://www.twoinchbrush.com/images/painting288.png	Autumn Mountain
289	https://www.twoinchbrush.com/images/painting289.png	Peaceful Valley
290	https://www.twoinchbrush.com/images/painting290.png	Seascape
291	https://www.twoinchbrush.com/images/painting291.png	Mountain Lake

1-10 of 403 rows | 1-3 of 27 columns Previous **1** 2 3 4 5 6 … 41 Next

Make a model that predicts, for each episode number within a season, how many colors will be used. Your prediction for episode 1 should only depend on the data for episode 1 (across all the seasons) and should not be influenced by how many colors were used in other episodes, and the same for all other episode numbers. Plot your data and model.

```
episode_models <- br %>%
  group_by(episode) %>%
  do(model = lm(num_colors ~ 1, data = .))

predictions <- episode_models %>%
  rowwise() %>%
  mutate(predicted_colors = predict(model, newdata = data.frame(episode = episode))) %>%
  ungroup()

ggplot(br, aes(x = factor(episode), y = num_colors)) +
  geom_jitter(width = 0.2, alpha = 0.5) +
  geom_point(data = predictions, aes(y = predicted_colors), color = "red", size = 3) +
  labs(title = "Number of Colors Used in Bob Ross Episodes",
       x = "Episode Number",
       y = "Number of Colors") +
  theme_minimal()
```