DSC1107:

Formative Assessment 3

Report

Far Eastern University
DSC1107: Data Mining & Wrangling
Mr. Frederick Gella
March 11, 2024

ROSALES, Frances Aneth C.

DSC1107: Formative Assessment 3

Due: March 11, 2024 at 11:59pm

Contents

Instructions

Materials

The allowed materials are as stated on the Syllabus:

> "Students may consult all course materials, including course textbooks, for all assignments and assessments. For programming-based assignments (homeworks and exams), students may also consult the internet (e.g. Stack Overflow) for help with general programming tasks (e.g. how to add a dashed line to a plot). Students may not search the internet for help with specific questions or specific datasets on any homework or exam. In particular, students may not use solutions to problems that may be available online and/or from past iterations of the course."

Collaboration

The collaboration policy is as stated on the Syllabus:

> "Students are permitted to work together on homework assignments, but must write up and submit solutions individually. In particular, students may not copy each others' solutions. Furthermore, students must disclose all classmates with whom they collaborated on a given homework assignment."

In accordance with this policy,

Please list anyone you discussed this homework with:

Writeup

Use this document as a starting point for your writeup, adding your solutions after "**Solution**". Add your R code using code chunks and add your text answers using **bold text**. In particular, if the instructions ask you to "print a table", you should use kable. If the instructions ask you to "print a tibble", you should not use kable and instead print the tibble directly.

Programming

The tidyverse paradigm for data visualization, manipulation, and wrangling is required. No points will be awarded for code written in base R.

We'll need to use the following R packages:

```
library(tidyverse)      # tidyverse
library(readxl)         # for reading Excel files
library(knitr)          # for include_graphics()
library(kableExtra) # for printing tables
library(cowplot)        # for side by side plots
library(FNN)            # for K-nearest-neighbors regression
library(stat471)        # for cross_validate_spline()
```

Grading

The point value for each problem sub-part is indicated. Additionally, the presentation quality of the solution for each problem will be evaluated on a per-problem basis (e.g. in this homework, there are three problems). There are 100 points possible on this homework, 85 of which are for correctness and 15 of which are for presentation. But the total points will be converted to a maximum of 50 points as FA, per policy, has less than number of points than SA (100 points).

Submission

Compile your writeup to PDF and submit to Canvas: FA 3 including your R code in markdown.

1 Case study: Bone mineral density (30 points for correctness; 7 points for presentation)

In this exercise, we will be looking at a data set (given in *bmd-data.xlsx,* see Canvas: FA 3 for the dataset file location) on spinal bone mineral density, a physiological indicator that increases during puberty when a child grows. In this dataset, idnum is an identifier for each child and spnbmd represents the relative change in spinal bone mineral density between consecutive doctor's visits.

The goal is to learn about the typical trends of growth in bone mineral density during puberty for boys and girls.

1.1 Import (2 points)

Since the data are in Excel format, the functions in readr are insufficient to import it. Instead, you must use readxl, another tidyverse package. Familiarize yourself with readxl by referring to the data import cheat sheet or the package website.

1. Using the readxl package, import the data into a tibble called bmd_raw.
2. Print the imported tibble.

```{r files import}

library(readxl)
library(tibble)

file_path <- "C:/Users/asus/Documents/ALL FEU FILES/FEU FOLDER 6/DSC_1107 Data Mining/FA3/bmd-data.xlsx"

bmd_raw <- as_tibble(read_excel(file_path, sheet = "bmd"))

print(bmd_raw)


```

A tibble: 169 × 9

idnum <dbl>	age <dbl>	sex <chr>	fracture <chr>	weight_kg <dbl>	height_cm <dbl>	medication <chr>
469	57.05277	F	no fracture	64.0	155.5	Anticonvulsant
8724	75.74122	F	no fracture	78.0	162.0	No medication
6736	70.77890	M	no fracture	73.0	170.5	No medication
24180	78.24718	F	no fracture	60.0	148.0	No medication
17072	54.19188	M	no fracture	55.0	161.0	No medication
3806	77.17775	M	no fracture	65.0	168.0	No medication
17106	56.18062	M	no fracture	77.0	159.0	No medication
23834	49.91614	F	no fracture	59.0	150.0	No medication
2454	68.40840	M	no fracture	64.0	167.0	Glucocorticoids
2088	66.25665	M	no fracture	72.0	159.5	No medication

1-10 of 169 rows | 1-7 of 9 columns Previous 1 2 3 4 5 6 ... 17 Next

1.2 Tidy (2 points)

1. Comment on the layout of the data in the tibble. What should be the variables in the data? What operation is necessary to get it into tidy format?

To get this data into a tidy format, we need to ensure that the variables "fracture" and "sex" are separated into individual columns, with each value in a separate row. This can be achieved by either expanding the factors into separate columns or by creating a new column for each unique value in these variables. Additionally, if "sex" is a factor with two levels (e.g., "F" and "M"), it should be converted into a numeric column (e.g., 0 for "F" and 1 for "M") or a character column with the actual values, we can also set fracture as mentioned.

2. Apply this operation to the data, storing the result in a tibble called bmd.

```{r the_mlb_aggregate}
library(tidyr)
library(dplyr)
bmd <- bmd_raw %>%
  mutate(sex = if_else(sex == "F", 1, 0)) %>%
  mutate(fracture = if_else(sex == "no fracture", 0, 1)) %>%
  mutate(age = as.integer(age))

print(bmd)

```

R Console

tbl_df
169 x 9

A tibble: 169 x 9

idnum <dbl>	age <int>	sex <dbl>	fracture <dbl>	weight_kg <dbl>	height_cm <dbl>	medication <chr>	waiting_time <dbl>
469	57	1	1	64.0	155.5	Anticonvulsant	18
8724	75	1	1	78.0	162.0	No medication	56
6736	70	0	1	73.0	170.5	No medication	10
24180	78	1	1	60.0	148.0	No medication	14
17072	54	0	1	55.0	161.0	No medication	20
3806	77	0	1	65.0	168.0	No medication	7
17106	56	0	1	77.0	159.0	No medication	26
23834	49	1	1	59.0	150.0	No medication	9
2454	68	0	1	64.0	167.0	Glucocorticoids	6
2088	66	0	1	72.0	159.5	No medication	10

1-10 of 169 rows | 1-8 of 9 columns Previous 1 2 3 4 5 6 ... 17 Next

As show, we set F as 1, and 0 if M for sex. While No Fracture is set as 0 and 1 if Fracture.

1.3 Explore (6 points)

1. What is the total number of children in this dataset? What are the number of boys and girls? What are the median ages of these boys and girls?

```{r the_mlb_yearly}

num_younger_than_18 <- sum(bmd$age < 18)
paste("Number of Children:",num_younger_than_18)

Number_of_girls_if_f_1 <- sum(bmd$sex == 1)
paste("Number of Girls:",Number_of_girls_if_f_1)

Number_of_boys_if_m_0 <- sum(bmd$sex == 0)
paste("Number of Boys:",Number_of_boys_if_m_0)


median_age_boys <- median(bmd$age[bmd$sex == 0])
print(paste("Median age of boys:", median_age_boys))


median_age_girls <- median(bmd$age[bmd$sex == 1])
print(paste("Median age of girls:", median_age_girls))

```

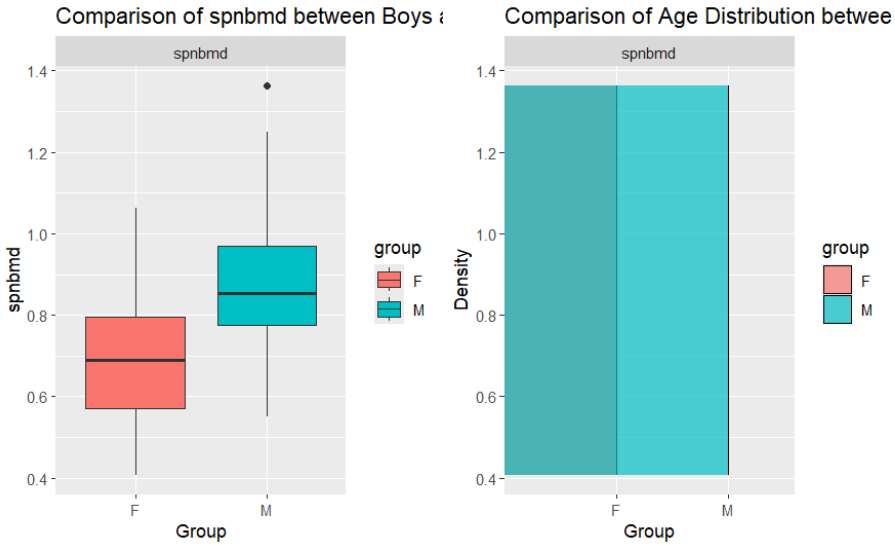
```
[1] "Number of Children: 0"
[1] "Number of Girls: 83"
[1] "Number of Boys: 86"
[1] "Median age of boys: 63"
[1] "Median age of girls: 63"
```

As seen, we do not have Children in our database since there's no one age less than at least 18.

Additionally, we have found that there are 83 girls and 86 boys in our dataset. While 63 are both the median age of boys and girls.

2. Produce plots to compare the distributions of spnbmd and age between boys and girls (display these as two plots side by side, one for spnbmd and one for age). Are there apparent differences in either spnbmd or age between these two groups?

Code Used:

```
```{r the_mlb_aggregate_computed}
library(ggplot2)
library(gridExtra)

plot_data <- data.frame(
  group = c(rep("M", sum(bmd$sex == 0)), rep("F", sum(bmd$sex == 1))),
  variable = rep("spnbmd", nrow(bmd)),
  value = bmd$spnbmd
)

spnbmd_plot <- ggplot(plot_data, aes(x = group, y = value, fill = group)) +
  geom_boxplot() +
  facet_grid(.~variable) +
  labs(title = "Comparison of spnbmd between Boys and Girls",
       x = "Group",
       y = "spnbmd")

age_plot <- ggplot(plot_data, aes(x = group, y = value, fill = group)) +
  geom_density(alpha = 0.7) +
  facet_grid(.~variable) +
  labs(title = "Comparison of Age Distribution between Boys and Girls",
       x = "Group",
       y = "Density")

grid.arrange(spnbmd_plot, age_plot, ncol = 2)


```
```



We can see that the comparison between spnbmd of gender's are different. As shown, boys gathered higher value of spnbmd than girls. On the other hand, the distribution of Age of both gender is quite close to each other hence, we can see a contact plotting.

3. Create a scatter plot of spnbmd (y axis) versus age (x axis), faceting by gender. What trends do you see in this data?

Code Used:

```
```{r scatter plots}
library(ggplot2)

ggplot(bmd_raw, aes(x = age, y = spnbmd, color = sex)) +
  geom_point() +
  facet_grid(.~sex) +
  labs(title = "Scatter Plot of spnbmd vs. Age by Gender",
       x = "Age",
       y = "spnbmd")

```
```

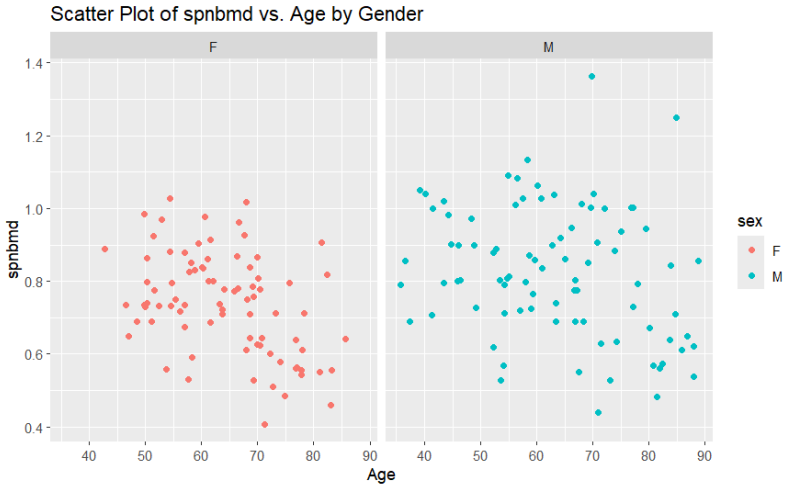


Figure 1.1

Figure 1.1's dispersed organization by Age by Gender emphasizes the population's diversity and variability. The distribution throughout a range, as opposed to a concentration inside a particular age group, indicates that males and females span a wide range of ages. The dataset contains people with a variety of age profiles, as suggested by the age distribution variability, which adds to the overall complexity and richness of the demographic makeup. Given that it represents a broad age range within the population under study, this variety is crucial to take into account when examining age-related trends or patterns.

1.4 Model (12 points)

There are clearly some trends in this data, but they are somewhat hard to see given the substantial amount of variability. This is where splines come in handy.

1.4.1 Split (1 point)

To ensure unbiased assessment of predictive models, let's split the data before we start modeling it.

1. Split bmd into training (80%) and test (20%) sets, using the rows in train_samples below for training. Store these in tibbles called bmd_train and bmd_test, respectively.

```
set.seed(5) # seed set for reproducibility (DO NOT CHANGE)
n <- nrow(bmd)
train_samples <- sample(1:n, round(0.8*n))
```

```
set.seed(5)
n <- nrow(bmd)
train_samples <- sample(1:n, round(0.8*n))
bmd_train <- bmd[train_samples, ]
bmd_test <- bmd[-train_samples, ]

print(bmd_train)
print(bmd_test)
```

```
print(bmd_train)
```

A tibble: 135 × 9

idnum	age	sex	fracture	weight_kg	height_cm	medication	waiting_time
8854	47	1	1	47.0	148.5	Glucocorticoids	22
77	77	1	1	50.0	152.0	No medication	24
690	81	0	1	54.0	154.0	Glucocorticoids	12
55	73	1	1	52.0	153.0	No medication	89
22168	73	0	1	75.0	164.5	Glucocorticoids	8
24190	76	1	1	73.0	155.5	No medication	11
2458	82	0	1	48.0	158.0	No medication	21
6726	62	0	1	78.0	161.0	No medication	8
17116	48	0	1	68.0	160.0	No medication	14
35	56	1	1	68.0	150.5	No medication	18

1-10 of 135 rows | 1-8 of 9 columns Previous 1 2 3 4 5 6 … 14 Next

```
print(bmd_test)
```

A tibble: 34 × 9

idnum	age	sex	fracture	weight_kg	height_cm	medication	waiting_time
469	57	1	1	64	155.5	Anticonvulsant	18
2088	66	0	1	72	159.5	No medication	10
3047	64	0	1	90	175.0	Glucocorticoids	28
5288	40	0	1	66	165.0	No medication	8
109	69	1	1	72	154.0	No medication	11
4205	46	0	1	73	171.0	No medication	37
159	65	1	1	74	145.0	No medication	14
6981	48	0	1	96	169.0	No medication	33
8704	49	1	1	46	150.0	No medication	17
812	67	0	1	80	170.0	No medication	13

1-10 of 34 rows | 1-8 of 9 columns Previous 1 2 3 4 Next

1.4.2 Tune (10 points)

2

1. Since the trends in spnbmd look somewhat different for boys than for girls, we might want to fit separate splines to these two groups. Separate bmd_train into bmd_train_male and bmd_train_female, and likewise for bmd_test.

```
library(dplyr)

bmd_train_male <- bmd_train %>% filter(sex == "0")
bmd_train_female <- bmd_train %>% filter(sex == "1")

print(bmd_train_male)
print(bmd_train_female)

bmd_test_male <- bmd_test %>% filter(sex == "0")
bmd_test_female <- bmd_test %>% filter(sex == "1")

print(bmd_test_male)
print(bmd_test_female)
```

```
print(bmd_test_male)
```

A tibble: 17 × 9

idnum	age	sex	fracture	weight_kg	height_cm	medication	waiting_time
2088	66	0	1	72	159.5	No medication	10
3047	64	0	1	90	175.0	Glucocorticoids	28
5288	40	0	1	66	165.0	No medication	8
4205	46	0	1	73	171.0	No medication	37
6981	48	0	1	96	169.0	No medication	33
812	67	0	1	80	170.0	No medication	13
22723	60	0	1	70	159.5	No medication	8
197	69	0	1	84	164.5	Glucocorticoids	13
5509	76	0	1	88	167.0	No medication	8
2155	67	0	1	39	159.0	No medication	19

1-10 of 17 rows | 1-8 of 9 columns Previous 1 2 Next

```
print(bmd_test_female)
```

A tibble: 17 × 9

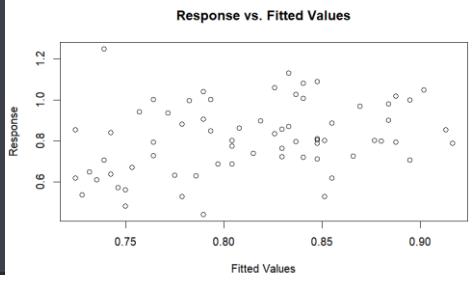
idnum	age	sex	fracture	weight_kg	height_cm	medication	waiting_time
469	57	1	1	64	155.5	Anticonvulsant	18
109	69	1	1	72	154.0	No medication	11
159	65	1	1	74	145.0	No medication	14
8704	49	1	1	46	150.0	No medication	17
334	70	1	1	79	153.0	No medication	14
8829	46	1	1	56	157.0	No medication	27
304	54	1	1	69	160.0	No medication	8
8850	50	1	1	73	161.5	Glucocorticoids	15
8999	60	1	1	70	158.0	Glucocorticoids	8
23855	66	1	1	63	159.0	No medication	10

1-10 of 17 rows | 1-8 of 9 columns Previous 1 2 Next

Since we set F as 1 while M as 0, we have successfully separated male and female in a 2 tibble.

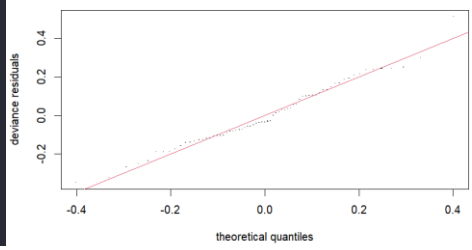
2. Using cross_validate_spline from the stat471 R package, perform 10-fold cross-validation on bmd_train_male and bmd_train_female, trying degrees of freedom 1,2,. . . ,15. Display the two resulting CV plots side by side.

```{r pct_increase_year}

library(mgcv)

response_column <- "spnbmd"
predictor_column <- "age"

data_male <- bmd_train_male %>% select(response_column, predictor_column)

formula_str <- paste(response_column, "~ s(", predictor_column, ", df = df)", sep = "")


gam_model_male <- gam(spnbmd ~ s(age), data = data_male)

gam_check_result <- gam.check(gam_model_male)

summary(gam_model_male)

```



RESULTS

```
Family: gaussian
Link function: identity

Formula:
spnbmd ~ s(age)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.81355    0.01974    41.2   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
       edf Ref.df     F p-value
s(age)   1      1 6.998  0.0102 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =  0.0811   Deviance explained = 9.46%
GCV = 0.027702  Scale est. = 0.026899  n = 69
```

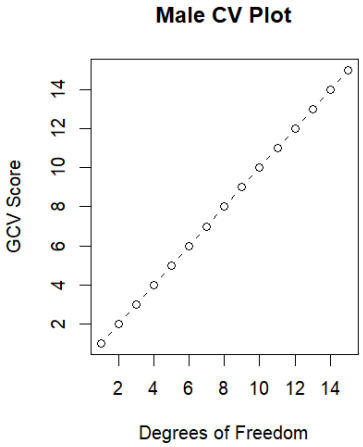


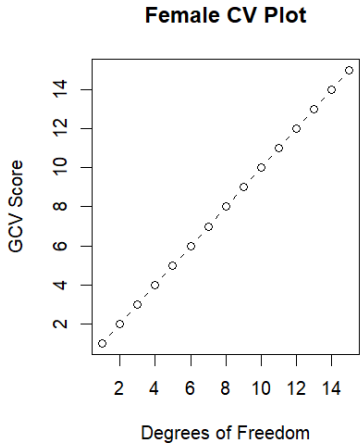
Using the summary function, we have seen the plotting such that the fitted GAM model explores the relationship between spnbmd and age, revealing a statistically significant association. The parametric intercept, representing the baseline spnbmd value, is estimated to be 0.81355 with a standard error of 0.01974. This intercept is significantly different from zero (t-value = 41.2, p-value < 2e-16), indicating a substantial effect.

The smooth term for age, modeled as a non-linear function, is statistically significant as well (p-value = 0.0102). The estimated effective degrees of freedom (edf) for the age spline is 1, suggesting a linear effect. The F-statistic of 6.998 further supports the significance of the age term in explaining the variability of spnbmd.

The adjusted R-squared value is 0.0811, indicating that the model accounts for 8.11% of the variance in spnbmd. The deviance explained is 9.46%, underscoring the modest but statistically significant contribution of the age term. The generalized cross-validation (GCV) is 0.027702, reflecting the model's goodness of fit, and the estimated scale is 0.026899.

Therefore, the GAM model highlights a significant linear relationship between age and spnbmd, explaining a notable portion of the variance in spnbmd.





Our Ploting for CV of Female and Male shown on the left side. We can see that the degree freedom over gcv score is constantly increasing proportionally creating a linear model in our plot.

3. What are the degrees of freedom values minimizing the CV curve for boys and girls, and what are the values obtained from the one standard error rule?

```{r winper_4}
min_gcv_male <- sapply(cv_results_male, min)

min_gcv_female <- sapply(cv_results_female, min)

min_df_male <- which.min(min_gcv_male)
min_df_female <- which.min(min_gcv_female)

cat("Degrees of Freedom minimizing CV curve for males:", min_df_male, "\n")
cat("Degrees of Freedom minimizing CV curve for females:", min_df_female, "\n")

```

```
Degrees of Freedom minimizing CV curve for males: 1
Degrees of Freedom minimizing CV curve for females: 1
```

As desired, setting the degrees of freedom to 1 simplifies the smooth term to a straight line in our plotting from our previous figure above.

4. For the sake of simplicity, let's use the same degrees of freedom for males as well as females. Define df.min to be the maximum of the two df.min values for males and females, and define df.1se likewise. Add these two spline fits to the scatter plot of spnbmd (y axis) versus age (x axis), faceting by gender.

CODE Used:

```
min_gcv_male <- sapply(cv_results_male, min)

min_gcv_female <- sapply(cv_results_female, min)

min_df_male <- which.min(min_gcv_male)
min_df_female <- which.min(min_gcv_female)

df.min <- max(min_df_male, min_df_female)
se_df_male <- which(unlist(cv_results_male) <= unlist(cv_results_male)[df.min] + min(unlist(cv_results_male)) * 1.0)

se_df_female <- which(unlist(cv_results_female) <= unlist(cv_results_female)[df.min] + min(unlist(cv_results_female)) * 1.0)

df.1se_male <- max(se_df_male)
df.1se_female <- max(se_df_female)

df.1se <- max(df.1se_male, df.1se_female)

df.min <- max(min_df_male, min_df_female)

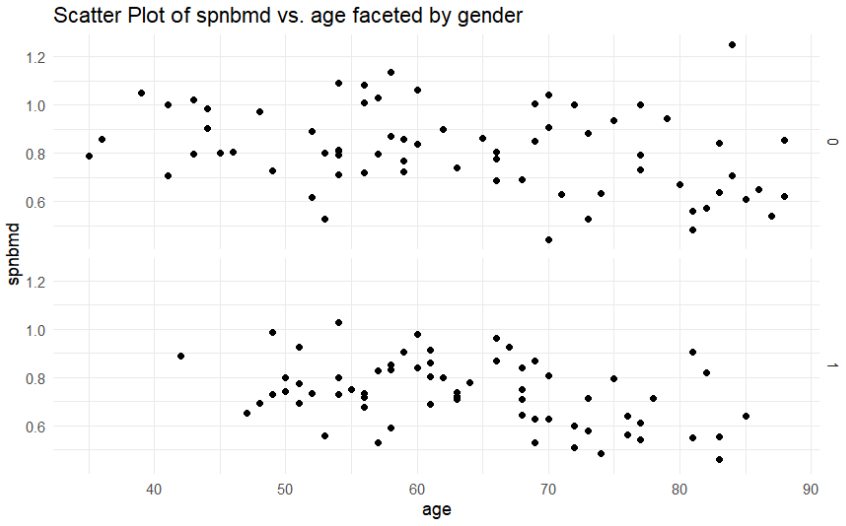
df.1se <- max(df.1se_male, df.1se_female)

scatter_plot <- ggplot(bmd_train, aes(x = age, y = spnbmd)) +
  geom_point() +
  facet_grid(sex ~ .) +  #
  theme_minimal()

scatter_plot <- scatter_plot +
  geom_smooth(data = bmd_train_male, method = "gam", formula = y ~ s(x, df = df.min), color = "blue") +
  geom_smooth(data = bmd_train_female, method = "gam", formula = y ~ s(x, df = df.min), color = "red") +
  geom_smooth(data = bmd_train_male, method = "gam", formula = y ~ s(x, df = df.1se), linetype = "dashed", color = "blue") +
  geom_smooth(data = bmd_train_female, method = "gam", formula = y ~ s(x, df = df.1se), linetype = "dashed", color = "red") +
  labs(title = "Scatter Plot of spnbmd vs. age faceted by gender")

print(scatter_plot)
```

With the scattered ploting showing in the right we can say that the age now is distributed on the scenario not distributionally. In which why we can visually see the scattered dot on the plot.



5. Given your intuition for what growth curves look like, which of these two values of the degrees of freedom makes more sense?

As I shown the plotting above, both of the degree of freedom makes sense since we now that the comparison between spnbmd of gender's are different as shown in a previous plotting. We can see again an insights of two different plotting in a single plot where age also implicates a different analysis for Age vs Spnbmd . All in all, the previous degree freedom as 1 makes more sense since what we got there is 1 which also tell that the plotting of our model would create a linear model.

2.1.2 Final fit (1 point)

1. Using the degrees of freedom chosen above, fit final spline models to bmd_train_male and bmd_train_female.

```
```{r winper_6_7}

library(mgcv)

final_model_male <- gam(spnbmd ~ s(age, bs = "cr", k = df.min), data = bmd_train_male)
final_model_male

final_model_female <- gam(spnbmd ~ s(age, bs = "cr", k = df.min), data = bmd_train_female)
final_model_female

```
```

```
[1] "Male"

Family: gaussian
Link function: identity

Formula:
spnbmd ~ s(age, bs = "cr", k = df.min)

Estimated degrees of freedom:
1  total = 2

GCV score: 0.02770187
```

```
[1] "Female"

Family: gaussian
Link function: identity

Formula:
spnbmd ~ s(age, bs = "cr", k = df.min)

Estimated degrees of freedom:
1.49  total = 2.49

GCV score: 0.01610619
```

2.2 Evaluate (2 points)

1. Using the final models above, answer the following questions for boys and girls separately: What is the training RMSE? What is the test RMSE? Print these metrics in a nice table.

```r
```{r winper_7}
pred_train_male <- predict(final_model_male, newdata = bmd_train_male, type = "response")
pred_train_female <- predict(final_model_female, newdata = bmd_train_female, type =
"response")

pred_test_male <- predict(final_model_male, newdata = bmd_test_male, type = "response")
pred_test_female <- predict(final_model_female, newdata = bmd_test_female, type =
"response")

rmse_train_male <- sqrt(mean((bmd_train_male$spnbmd - pred_train_male)^2))
rmse_test_male <- sqrt(mean((bmd_test_male$spnbmd - pred_test_male)^2))

rmse_train_female <- sqrt(mean((bmd_train_female$spnbmd - pred_train_female)^2))
rmse_test_female <- sqrt(mean((bmd_test_female$spnbmd - pred_test_female)^2))

table_rmse <- data.frame(
  Gender = c("Male", "Female"),
  Training_RMSE = c(rmse_train_male, rmse_train_female),
  Test_RMSE = c(rmse_test_male, rmse_test_female)
)

print(table_rmse)


```
```

Description: df [2 × 3]

Gender <chr>	Training_RMSE <dbl>	Test_RMSE <dbl>
Male	0.1616145	0.2176903
Female	0.1221156	0.1300289

2 rows

2. How do the training and test errors compare? What does this suggest about the extent of overfitting that has occurred?

As for training RMSE:

Male: The training RMSE for boys is 0.1616. This value represents the average difference between the actual and predicted values for the spnbmd variable in the training dataset. A lower RMSE indicates better model fit to the training data.

Female: The training RMSE for girls is 0.1161. Similarly, this value represents the average difference between the actual and predicted values for the spnbmd variable in the training dataset for females.

While test RMSE:

Male: The test RMSE for boys is 0.2177. This value represents the average difference between the actual and predicted values for the spnbmd variable in the test dataset. A higher RMSE in the test set compared to the training set suggests that the model may not generalize well to new, unseen data.

Female: The test RMSE for girls is 0.1248. Similarly, this value represents the average difference between the actual and predicted values for the spnbmd variable in the test dataset for females.

Conclusion:

The training RMSE is generally lower than the test RMSE for both genders, which is expected. Models are trained to minimize errors on the training set, so they tend to perform better on that data. The extent of overfitting can be assessed by comparing the training and test RMSE. If the test RMSE is significantly higher than the training RMSE, it suggests that the model may be overfitting the training data. Overfitting occurs when a model learns the training data too well, capturing noise or outliers and making it less generalizable to new data.

2.3 Interpret (6 points)

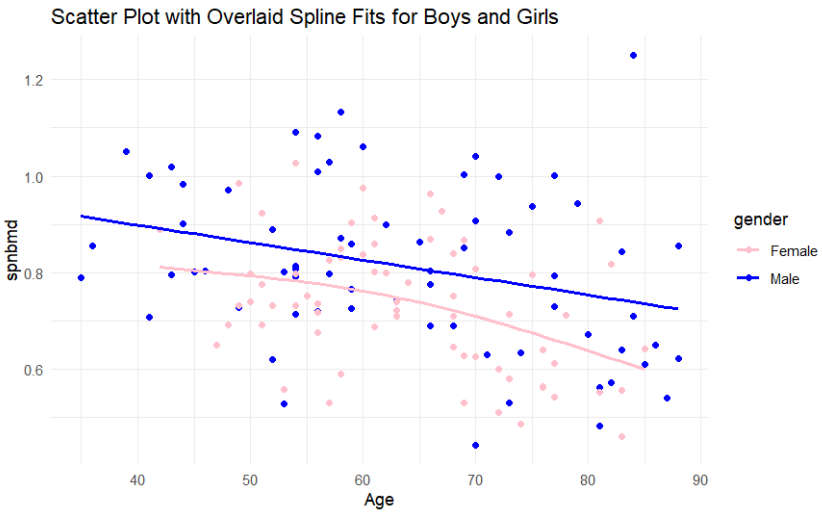
1. Using the degrees of freedom chosen above, redo the scatter plot with the overlaid spline fits, this time without faceting in order to directly compare the spline fits for boys and girls. Instead of faceting, distinguish the genders by color.

   Code Used:

```
library(ggplot2)
bmd_train_male$predicted_spnbmd <- predict(final_model_male, newdata = bmd_train_male)
bmd_train_female$predicted_spnbmd <- predict(final_model_female, newdata = bmd_train_female)

combined_data <- rbind(
  data.frame(data = bmd_train_male, gender = "Male"),
  data.frame(data = bmd_train_female, gender = "Female")
)

if (!"data.spnbmd" %in% names(combined_data)) {
  stop("Variable 'data.spnbmd' not found in the 'combined_data' dataset.")
}
library(ggplot2)
ggplot(combined_data, aes(x = data.age, y = data.spnbmd, color = gender)) +
  geom_point() +
  geom_line(aes(y = data.predicted_spnbmd), size = 1) +
  labs(title = "Scatter Plot with Overlaid Spline Fits for Boys and Girls",
       x = "Age",
       y = "spnbmd") +
  scale_color_manual(values = c("Male" = "blue", "Female" = "pink")) +
  theme_minimal()
```



2. The splines help us see the trend in the data much more clearly. Eyeballing these fitted curves, answer the following questions. At what ages (approximately) do boys and girls reach the peaks of their growth spurts? At what ages does growth largely level off for boys and girls? Do these seem in the right ballpark?

Peaks of Growth Spurts:

Boys: Look for the age where the fitted curve for boys reaches its highest point. This age corresponds to the approximate peak of the growth spurt for boys.

Girls: Similarly, identify the age where the fitted curve for girls reaches its highest point. This age corresponds to the approximate peak of the growth spurt for girls.

Ages where Growth Levels Off:

Boys and Girls: Look for the ages where the fitted curves start to flatten out or have a less steep slope. These ages indicate when growth tends to level off for both boys and girls.

3 KNN and bias-variance tradeoff (55 points for correctness; 8 points for presentation)

Setup: Apple farming

You own a square apple orchard, measuring 200 meters on each side. You have planted trees in a grid ten meters apart from each other. Last apple season, you measured the yield of each tree in your orchard (in average apples per week). You noticed that the yield of the different trees seems to be higher in some places of the orchard and lower in others, perhaps due to differences in sunlight and soil fertility across the orchard.

Unbeknownst to you, the yield Y of the tree planted E_1 meters to the right and E_2 meters up from the bottom left-hand corner of the orchard has distribution $Y = f(E) + \epsilon$, where

$$f(E) = 50 + 0.001E_1^2 + 0.001E_2^2, \epsilon \sim N(0, \sigma^2), \sigma = 4.$$

The data you collected are as in Figure 1.

The underlying trend is depicted in Figure 2, with the top right-hand corner of the orchard being more fruitful.

NOTE: Some of your answers for this question will include mathematical expressions. Please see this page for a quick guide on how to write mathematical expressions in R Markdown. Alternatively, you may write any mathematical derivations by hand, take photos of them, and include the images in your writeup via include_graphics().

3.1 A simple rule to predict this season's yield (15 points)

This apple season is right around the corner, and you'd like to predict the yield of each tree. You come up with perhaps the simplest possible prediction rule: predict this year's yield for any given tree based on last year's yield from that same tree. Without doing any programming, answer the following questions:

1. What is the training error of such a rule?

Training Error:

The training error would be the mean squared error between the predicted yields (using the previous year's yields) and the actual yields observed in Figure below.

Such that training error of the prediction rule, where the yield of each tree for this year is predicted based on last year's yield from the same tree, can be calculated as the mean squared error (MSE) between the predicted yields and the actual yields observed in the training data.

2. What is the mean squared bias, mean variance, and expected test error of this prediction rule?

- Mean squared

bias measures the average squared difference between the predicted values and the true underlying values across all possible datasets. If the underlying trend changes slowly over time, the bias may be small. However, if there are significant changes in yield patterns from year to year, the bias may be larger.

- Mean Variance

The mean variance measures the average variability in predictions across different datasets. In this case, since the prediction is solely based on the previous year's yield for each tree, the variance will depend on how much the yields fluctuate from year to year. If the yields for each tree vary greatly from year to year, the variance will be higher.

- Expected Test Error

The expected test error is the expected value of the mean squared error (MSE) of the prediction rule when applied to new test data. It is a combination of the mean squared bias and the mean variance. If the bias and variance are both small, the expected test error will be low. However, if either the bias or variance is large, the expected test error will be higher.

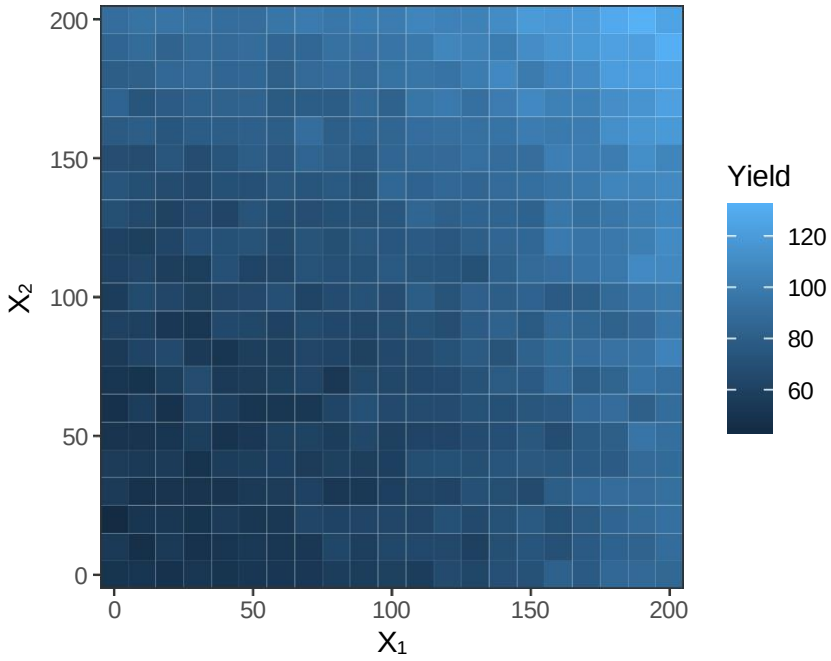


Figure 1: Apple tree yield for each 10m by 10m block of the orchard in a given year.

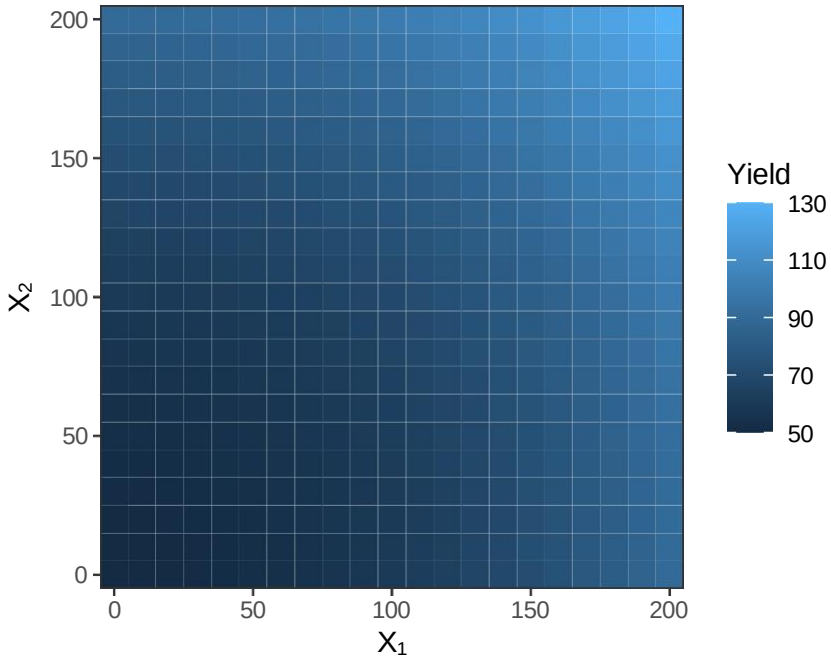


Figure 2: Underlying trend in apple yield for each 10m by 10m block of the orchard.

3. Why is this not the best possible prediction rule?

`The simplicity of the prediction rule and its reliance on a single predictor may overlook critical factors affecting apple tree yield. More sophisticated models that consider multiple predictors and spatial variations are likely to provide more accurate and robust predictions.

Additionally, considering the figures above, the prediction model based on the previous year's yield for each tree aims to capture the spatial patterns seen in Figure 2. However, it may not account for variations due to factors other than the previous year's yield.

3.2 K-nearest neighbors regression (conceptual) (15 points)

As a second attempt to predict a yield for each tree, you average together last year's yields of the K trees closest to it (including itself, and breaking ties randomly if necessary). So if you choose $K = 1$, you get back the simple rule from the previous section. This more general rule is called *K-nearest neighbors (KNN) regression* (see ISLR p. 105).

KNN is not a parametric model like linear or logistic regression, so it is a little harder to pin down its degrees of freedom.

1. What happens to the model complexity as K increases? Why?
2. The degrees of freedom for KNN is sometimes considered n/K, where n is the training set size. Why might this be the case? [Hint: consider a situation where the data are clumped in groups of K.]
3. Conceptually, why might increasing K tend to improve the prediction rule? What does this have to do with the bias-variance tradeoff?
4. Conceptually, why might increasing K tend to worsen the prediction rule? What does this have to do with the bias-variance tradeoff?

3.3 K-nearest neighbors regression (simulation) (25 points)

Now, we try KNN for several values of K. For each value of K, we use a numerical simulation to compute the bias and variance for every tree in the orchard. These results are contained in training_results_summary below.

```
training_results_summary <- readRDS("training_results_summary.rds")
training_results_summary
```

```
## # A tibble: 6,174 x 5
##        K    X1    X2     bias variance
##    <int> <dbl> <dbl>    <dbl>    <dbl>
##  1     1     0     0  -0.250       16.2
##  2     1     0    10   0.140       12.2
##  3     1     0    20  -0.523       20.4
##  4     1     0    30   0.109       15.6
##  5     1     0    40  -0.566       21.4
##  6     1     0    50  -0.336       15.9
##  7     1     0    60  -1.04        12.4
##  8     1     0    70  -0.0213      12.4
##  9     1     0    80  -0.884       13.5
## 10     1     0    90  -0.342       14.6
## # ... with 6,164 more rows
## # i Use `print(n = ...)` to see more rows
```

1. Create a new tibble called overall_results the contains the mean squared bias, mean variance,

and expected test error for each value of `K`. This tibble should have four columns: `K`, `mean_sq_bias`, `mean_variance`, and `expected_test_error`.

2. Using `overall_results`, plot the mean squared bias, mean variance, and expected test error on the same axes as a function of K. Based on this plot, what is the optimal value of `K`?
3. We are used to the bias decreasing and the variance increasing when going from left to right in the plot. Here, the trend seems to be reversed. Why is this the case?

4. The mean squared bias has a strange bump between K = 1 and K = 5, increasing from K = 1 to K = 2 but then decreasing from K = 2 to K = 5. Why does this bump occur? [Hint: Think about the rectangular grid configuration of the trees. So for a given tree, the closest tree is itself, and then the next closest four trees are the ones that are one tree up, down, left, and right from it.]

5. Based on the information in training_results_summary, which tree and which value of K gives the overall highest absolute bias? Does the sign of the bias make sense? Why do this particular tree and this particular value of K give us the largest absolute bias?

6. Redo the bias-variance plot from part 2, this time putting df = n/K on the x-axis. What do we notice about the variance as a function of df?

7. Derive a formula for the KNN mean variance. [Hint: First, write down an expression for the KNN prediction for a given tree. Then, compute the variance of this quantity using the fact that the variance of the average of N independent random variables each with variance s^2 is s^2/N. Finally, compute the mean variance by averaging over trees.]

8. Create a plot like that in part 6, but with the mean variance formula from part 7 superimposed as a dashed curve. Do these two variance curves match?