T02 Descriptive Statistics and Visualization

MATH 2411 Applied Statistics

WANG Zhiwei

MATH, HKUST

2024-09-16

Let's first simulate a dataset!

Student grading system

```
set.seed(20240916)
n_students <- 100; n_groups <- 5
each_group <- n_students / n_groups
id <- as.integer(c(1:n_students))
gender <- as.factor(sample(c("M", "F"), n_students, replace = TRUE))
group <- as.factor(rep(1:n_groups, each_group))
math <- round(runif(min = 55, max = 95, n = n_students))
english <- round(runif(min = 55, max = 95, n = n_students))
physics <- round(runif(min = 55, max = 95, n = n_students))
grading_system <- data.frame(id, group, gender, math, english, physics)
grading_system$average <- rowMeans(grading_system[, c("math", "english"
knitr::kable(head(grading_system, n = 3), format = 'html')
```

id	group	gender	math	english	physics	average
1	1	F	80	65	93	79.33333
2	2	M	86	65	73	74.66667
3	3	M	59	61	90	70.00000

Some simple descriptive statistics

```
dim(grading_system)
```

```
[1] 100 7
```

```
nrow(grading_system)
```

```
[1] 100
```

```
n_students
```

```
[1] 100
```

```
ncol(grading_system)
```

```
[1] 7
```

Some simple descriptive statistics

```
table(grading_system$gender)
```

F M 49 51

```
table(grading_system$group)
```

1 2 3 4 5 20 20 20 20 20

```
summary(grading_system$gender)
```

F M 49 51

```
summary(grading_system$math)
```

Min. 1st Qu. Median Mean 3rd Qu. Max. 57.00 64.75 77.00 75.89 86.00 94.00

Some simple descriptive statistics

```
knitr::kable(summary(grading_system[, c("math", "english", "physics", "a
```

	math	english	physics	average
	Min. :57.00	Min. :56.00	Min. :55.00	Min. :62.33
	1st Qu.:64.75	1st Qu.:65.00	1st Qu.:63.75	1st Qu.:70.67
	Median :77.00	Median :76.00	Median :76.00	Median :74.83
	Mean :75.89	Mean :75.79	Mean :75.05	Mean :75.58
	3rd Qu.:86.00	3rd Qu.:86.00	3rd Qu.:85.00	3rd Qu.:79.75
	Max. :94.00	Max. :95.00	Max. :95.00	Max. :88.33

Some simple descriptive statistics

```r
head(rowMeans(grading_system[, c("math", "english", "physics")]))
```

```
[1] 79.33333 74.66667 70.00000 83.66667 78.66667 64.66667
```

```r
colMeans(grading_system[, c("math", "english", "physics")])
```

```
math english physics 75.89 75.79 75.05
```

```r
colSums(grading_system[, c("math", "english", "physics")]) / nrow(gradi
```

```
math english physics 75.89 75.79 75.05
```

Missing value?

Set `na.rm = TRUE`.

Some simple descriptive statistics

```
var(grading_system[, "math"])
```

```
[1] 132.5029
```

```
mean((grading_system[, "math"] - mean(grading_system[, "math"]))^2)
```

```
[1] 131.1779
```

```
sum((grading_system[, "math"] - mean(grading_system[, "math"]))^2) / (nr
```

```
[1] 132.5029
```

```
sd(grading_system[, "math"])
```

```
[1] 11.51099
```

```
sqrt(var(grading_system[, "math"]))
```

```
[1] 11.51099
```

Some simple descriptive statistics

```
knitr::kable(var(grading_system[, c("math", "english", "physics")]), for
```

	math	english	physics
math	132.502929	5.128182	-7.620707
english	5.128182	146.329192	-24.928788
physics	-7.620707	-24.928788	147.805556

```
sum((grading_system[, "math"] - mean(grading_system[, "math"])) *
  (grading_system[, "physics"] - mean(grading_system[, "physics"]))) /
  (nrow(grading_system) - 1)
```

```
[1] -7.620707
```

Some simple descriptive statistics

```
knitr::kable(cor(grading_system[, c("math", "english", "physics")]), for
```

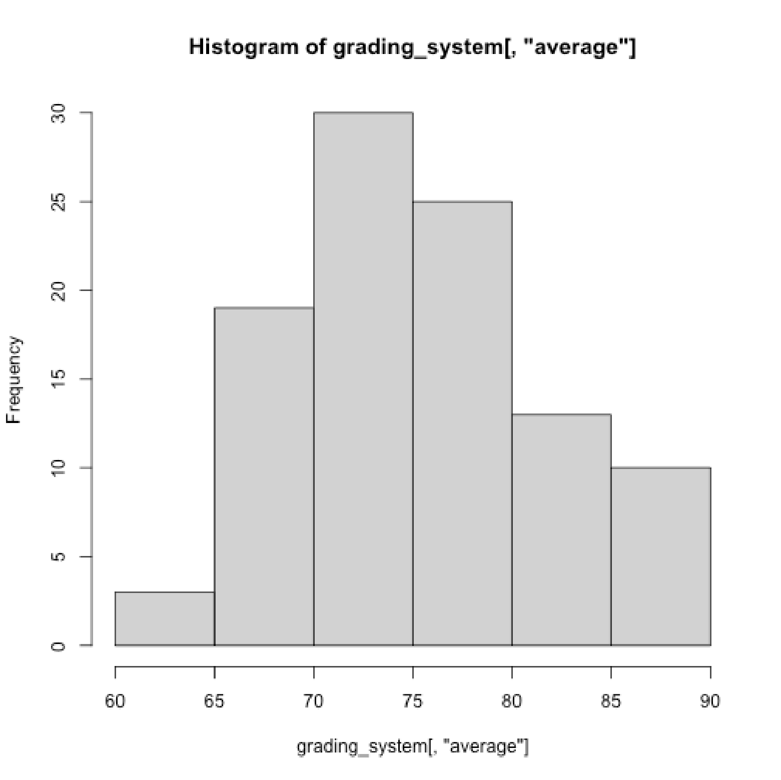
	math	english	physics
math	1.0000000	0.0368286	-0.0544549
english	0.0368286	1.0000000	-0.1695080
physics	-0.0544549	-0.1695080	1.0000000

In-class exercise: How to calculate the correlation between `math` and `physics` by yourself?

How about the visualization?

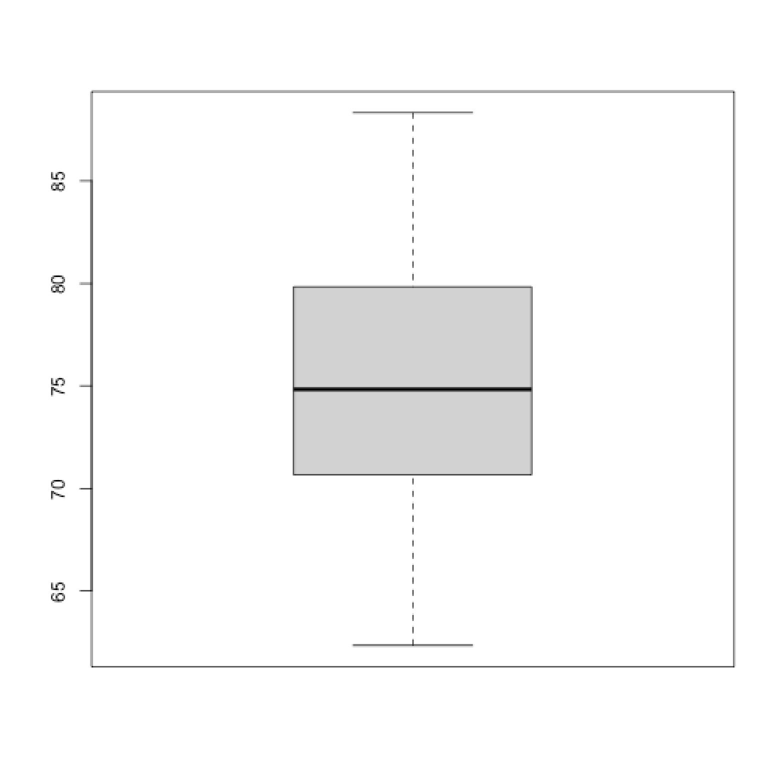
Histogram

```
hist(grading_system[, "average"])
```



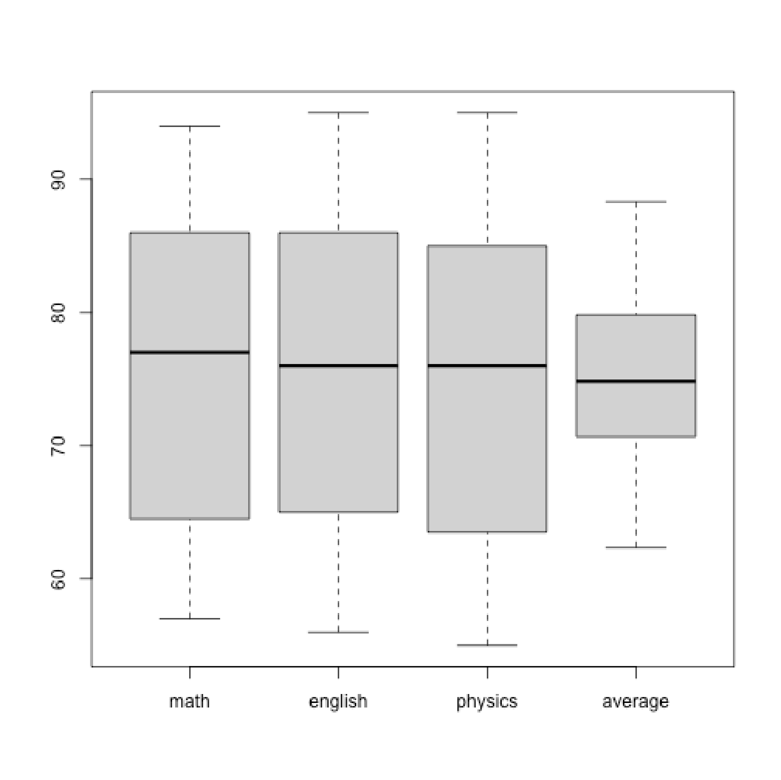
Box plot

```r
boxplot(grading_system[, "average"])
```



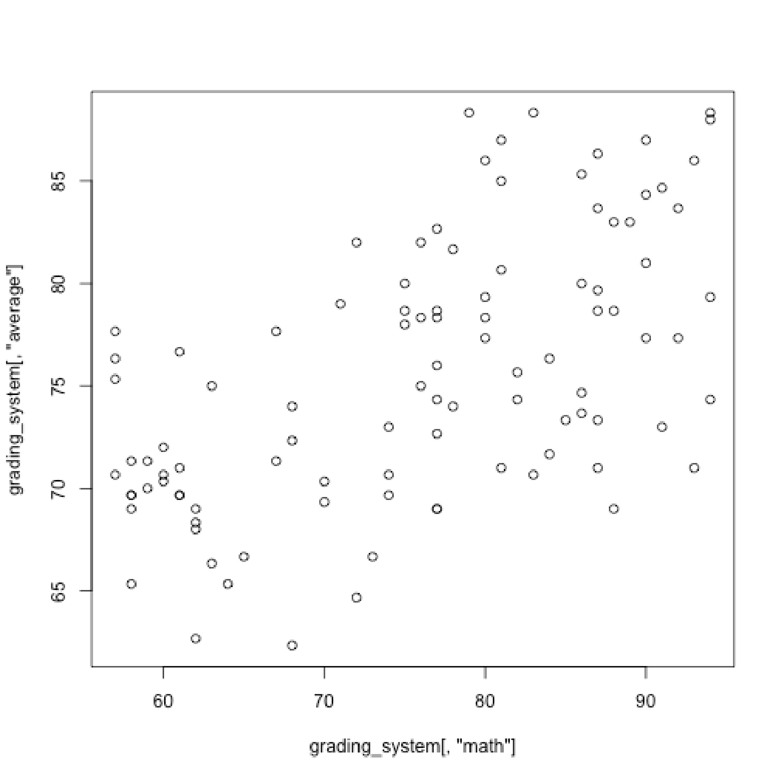
Box plot

```
boxplot(grading_system[, c("math", "english", "physics", "average")])
```



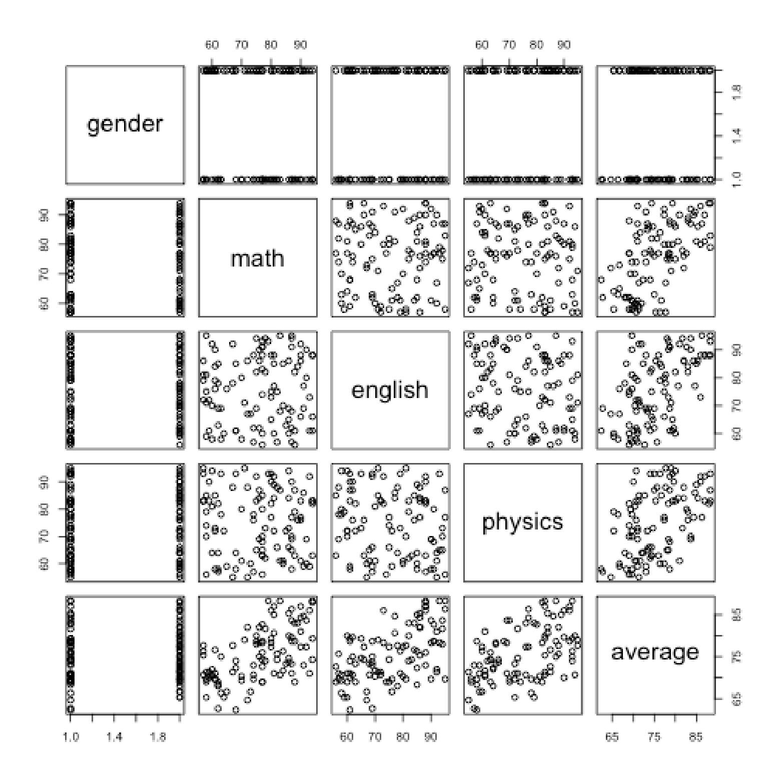
Scatter plot

```
plot(x = grading_system[, "math"], y = grading_system[, "average"])
```



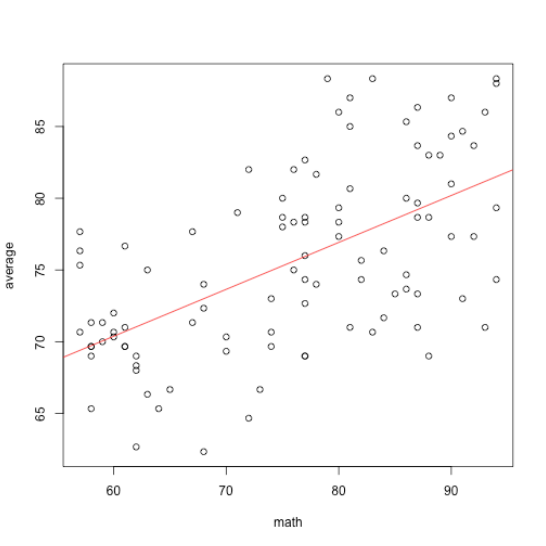
Scatter plot

```
plot(grading_system[, c("gender", "math", "english", "physics", "average
```



Scatter plot

```r
fit_lm <- lm(average ~ math, data = grading_system)
plot(average ~ math, data = grading_system)
abline(fit_lm, col = "red")
```



```r
fit_lm$coefficients
```

(Intercept) math 50.755858 0.327063

Want more beautiful figures?

Use ggplot2!

Someone said that if you don’t know ggplot2, it’s like you haven’t learned R!

Actually, "someone" is me:)

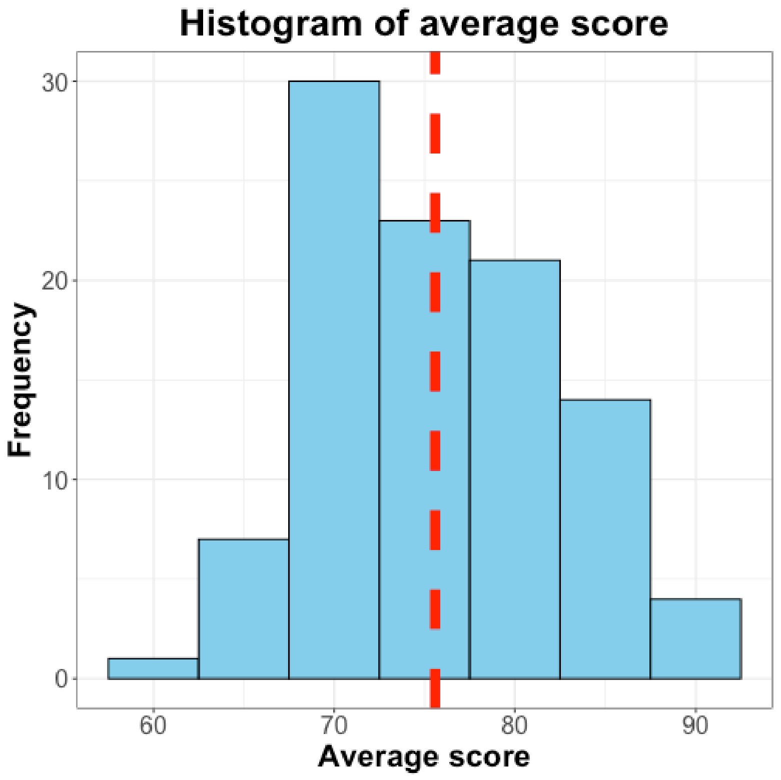
Histogram

Code Figure

```r
# install.packages("ggplot2")
library(ggplot2)
ggplot(grading_system, aes(x = average)) +
  geom_histogram(binwidth = 5, fill = "skyblue", color = "black") +
  geom_vline(aes(xintercept = mean(average)), color = "red", linetype =
  labs(title = "Histogram of average score", x = "Average score", y = "F
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 24, hjust = 0.5)
        axis.title = element_text(face = "bold", size = 20),
        axis.text = element_text(size = 12),
        axis.text.x = element_text(size = 16),
        axis.text.y = element_text(size = 16))
```

Histogram

Code Figure



Histogram in group

group argument Code Figure

We can also generate the histogram in group.

Histogram in group

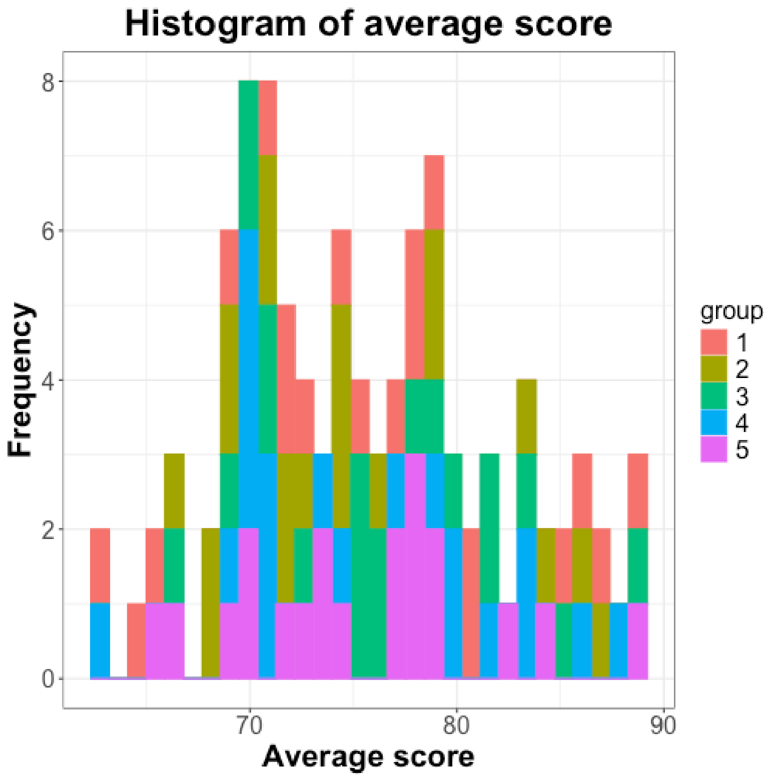
group argument   Code   Figure

```r
ggplot(grading_system, aes(x = average, group = group)) +
  geom_histogram(aes(fill = group, color = group)) +
  labs(title = "Histogram of average score", x = "Average score", y = "
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 24, hjust = 0.5)
        axis.title = element_text(face = "bold", size = 20),
        axis.text = element_text(size = 12),
        axis.text.x = element_text(size = 16),
        axis.text.y = element_text(size = 16),
        legend.title = element_text(size = 16),
        legend.text = element_text(size = 16))
```

Histogram in group

group argument Code Figure

```
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Histogram with `facet_grid()`

facet_grid() Code Figure

`facet_grid()` forms a matrix of panels defined by row and column faceting variables. It is most useful when you have two discrete variables, and all combinations of the variables exist in the data.

Let's see the following example to generate the histograms of all three subjects in each group.

Histogram with `facet_grid()`

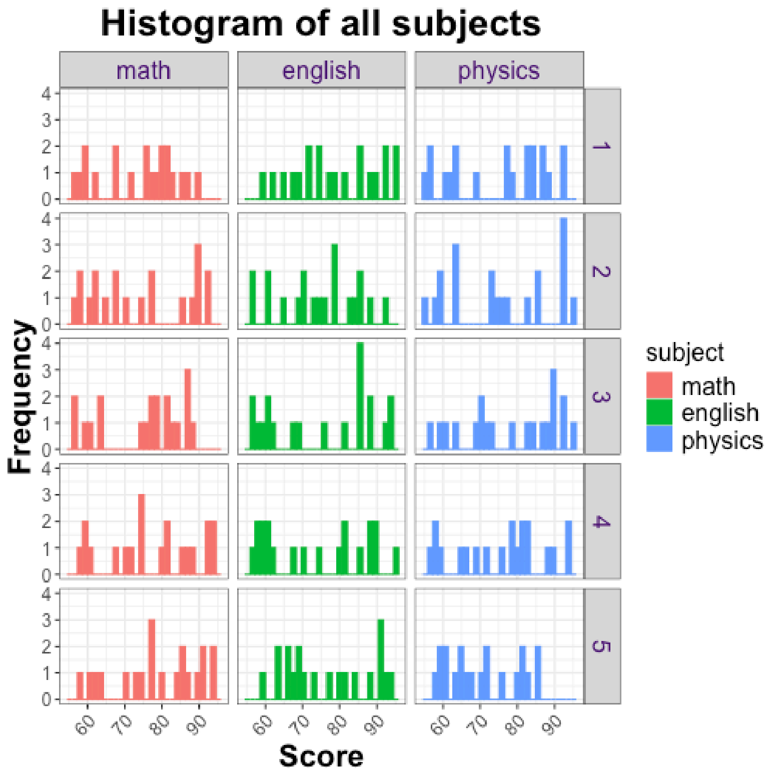
facet_grid()   Code   Figure

```r
grading_system_long <- reshape2::melt(grading_system[, -7],
                                      id = c("id", "group", "gender"),
                                      variable.name = "subject",
                                      value.name = "score")
ggplot(grading_system_long, aes(x = score)) +
  geom_histogram(aes(fill = subject, color = subject)) +
  labs(title = "Histogram of all subjects", x = "Score", y = "Frequency'
  facet_grid(group ~ subject) +
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 24, hjust = 0.5)
        axis.title = element_text(face = "bold", size = 20),
        axis.text = element_text(size = 16),
        axis.text.x = element_text(size = 12, angle = 45, hjust = 1),
        axis.text.y = element_text(size = 12),
        legend.title = element_text(size = 16),
        legend.text = element_text(size = 16),
        strip.text = element_text(size = 16, color = "#490573"))
```

Histogram with `facet_grid()`

facet_grid() Code Figure

```
#> `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Box plot

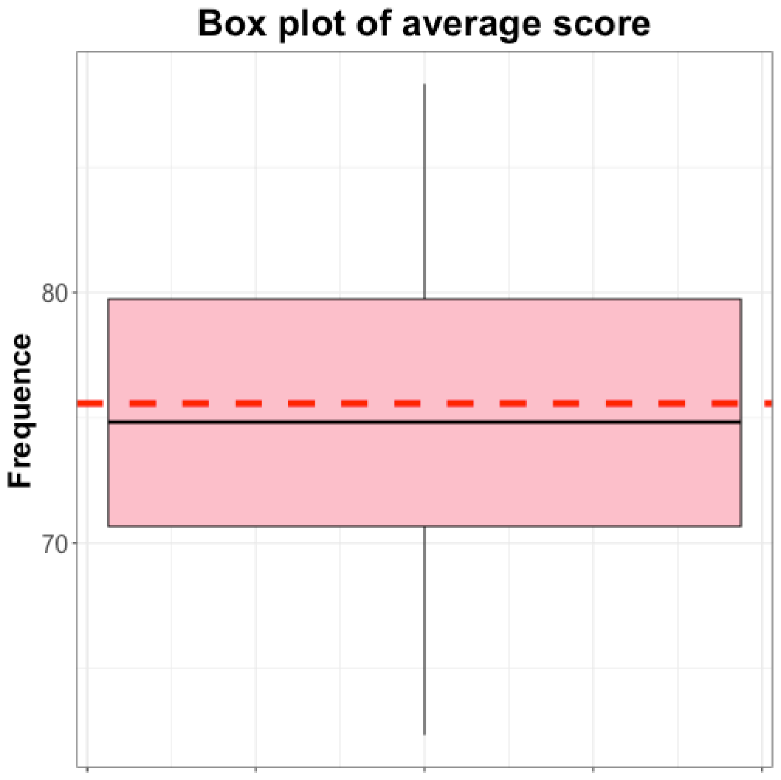
Code Figure

```r
ggplot(grading_system, aes(y = average)) +
  geom_boxplot(color = "black", fill = "pink") +
  geom_hline(aes(yintercept = mean(average)), color = "red", linetype =
  labs(title = "Box plot of average score", y = "Frequence") +
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 24, hjust = 0.5)
        axis.title = element_text(face = "bold", size = 20),
        axis.text = element_text(size = 12),
        axis.text.x = element_blank(),
        axis.text.y = element_text(size = 16))
```

Box plot

Code

Figure



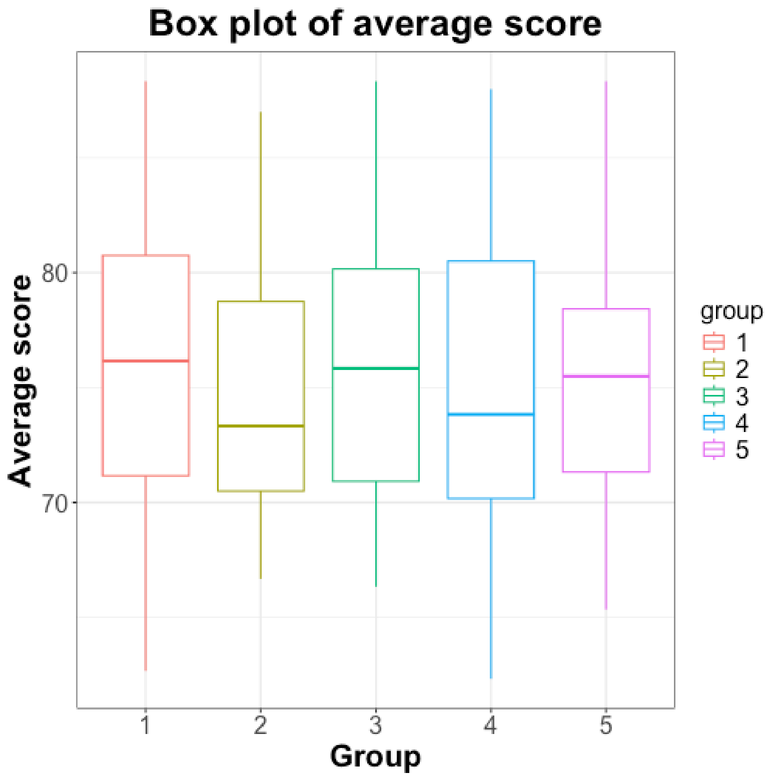
Box plot in group

Code Figure

```
ggplot(grading_system, aes(x = group, y = average)) +
  geom_boxplot(aes(color = group)) +
  labs(title = "Box plot of average score", x = "Group", y = "Average sc
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 24, hjust = 0.5)
        axis.title = element_text(face = "bold", size = 20),
        axis.text = element_text(size = 12),
        axis.text.x = element_text(size = 16),
        axis.text.y = element_text(size = 16),
        legend.title = element_text(size = 16),
        legend.text = element_text(size = 16))
```

Box plot in group

Code Figure



Box plots with `facet_grid()`

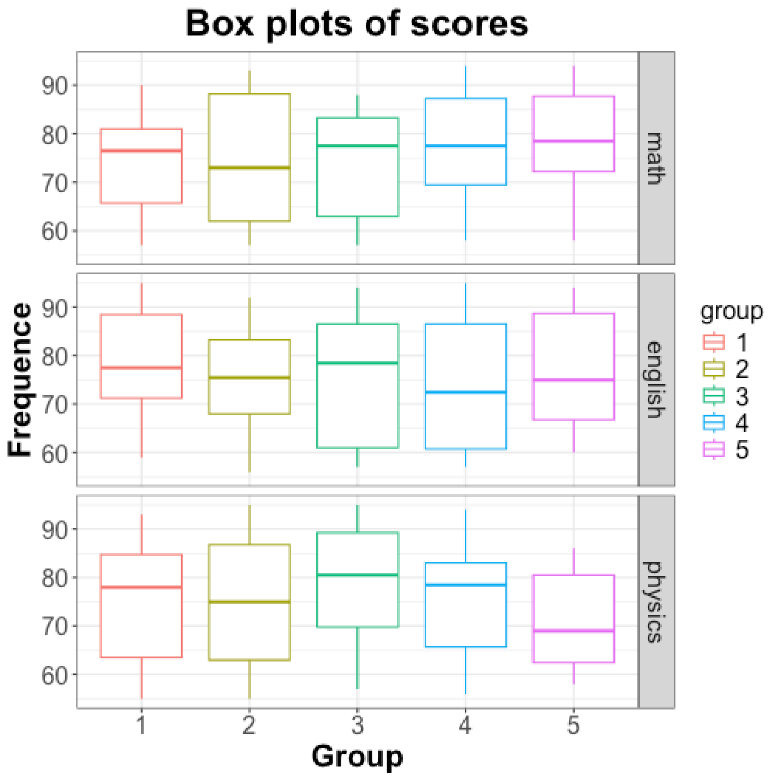
Code

Figure

```r
ggplot(grading_system_long, aes(x = group, y = score)) +
  geom_boxplot(aes(color = group)) +
  labs(title = "Box plots of scores", x = "Group", y = "Frequence") +
  facet_grid(subject ~ .) +
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 24, hjust = 0.5)
        axis.title = element_text(face = "bold", size = 20),
        axis.text = element_text(size = 16),
        axis.text.x = element_text(size = 16),
        axis.text.y = element_text(size = 16),
        legend.title = element_text(size = 16),
        legend.text = element_text(size = 16),
        strip.text = element_text(size = 16))
```

Box plots with `facet_grid()`

Code Figure



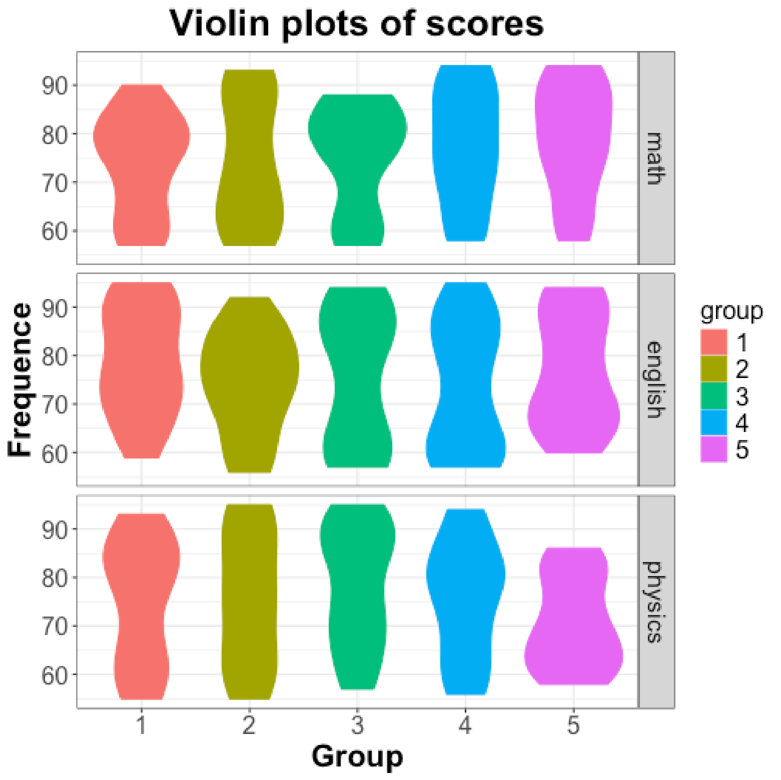
Violin plot

Code Figure

```
ggplot(grading_system_long, aes(x = group, y = score)) +
  geom_violin(aes(fill = group, color = group)) +
  labs(title = "Violin plots of scores", x = "Group", y = "Frequence") +
  facet_grid(subject ~ .) +
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 24, hjust = 0.5)
        axis.title = element_text(face = "bold", size = 20),
        axis.text = element_text(size = 16),
        axis.text.x = element_text(size = 16),
        axis.text.y = element_text(size = 16),
        legend.title = element_text(size = 16),
        legend.text = element_text(size = 16),
        strip.text = element_text(size = 16))
```

Violin plot

Code Figure



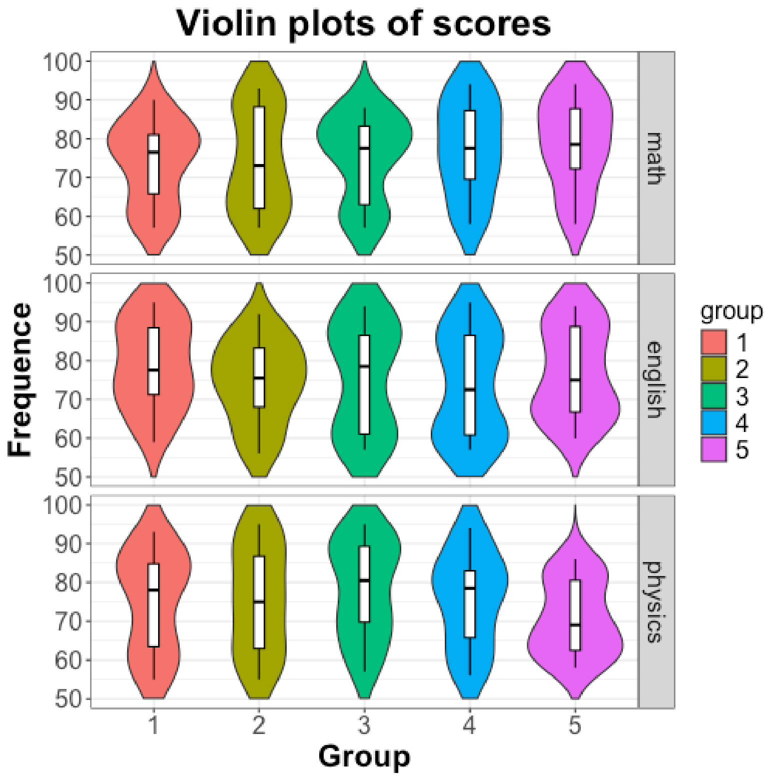
Violin plot with box plot inner

Code Figure

```
ggplot(grading_system_long, aes(x = group, y = score)) +
  geom_violin(aes(fill = group), trim = FALSE) +
  geom_boxplot(width = 0.1, fill = "white", color = "black") +
  ylim(50, 100) +
  labs(title = "Violin plots of scores", x = "Group", y = "Frequence") +
  facet_grid(subject ~ .) +
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 24, hjust = 0.5)
        axis.title = element_text(face = "bold", size = 20),
        axis.text = element_text(size = 16),
        axis.text.x = element_text(size = 16),
        axis.text.y = element_text(size = 16),
        legend.title = element_text(size = 16),
        legend.text = element_text(size = 16),
        strip.text = element_text(size = 16))
```

Violin plot with box plot inner

Code Figure



Scatter plot

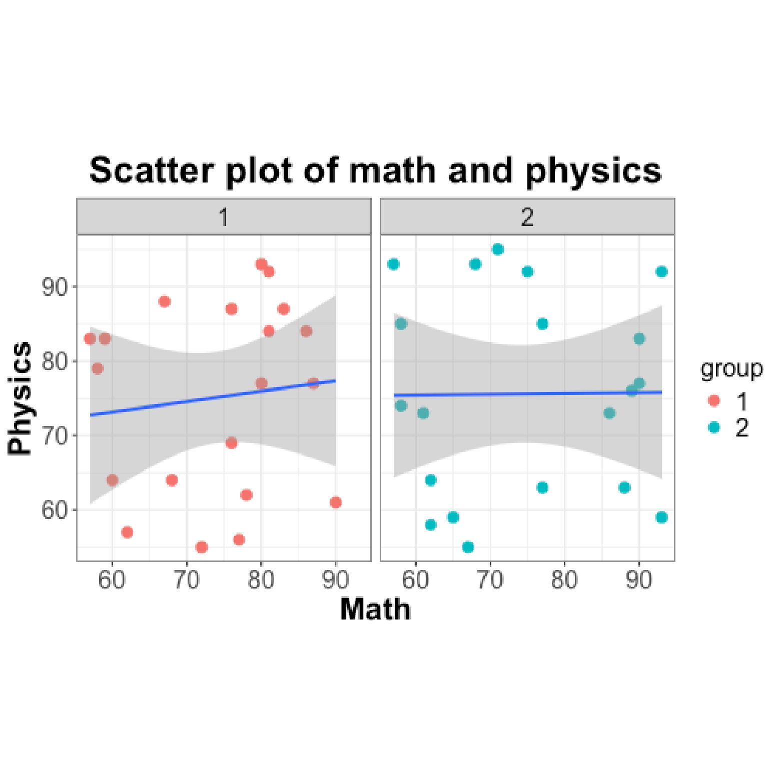
Code Figure

```r
# Use only first two groups
ggplot(grading_system[grading_system$group %in% c(1, 2), ], aes(x = math
  facet_grid(. ~ group) +
  geom_point(aes(color = group), size = 3) +
  geom_smooth(method = "lm") +
  labs(title = "Scatter plot of math and physics", x = "Math", y = "Phys
  coord_fixed(ratio = 1) +
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 24, hjust = 0.5)
        axis.title = element_text(face = "bold", size = 20),
        axis.text = element_text(size = 16),
        axis.text.x = element_text(size = 16),
        axis.text.y = element_text(size = 16),
        legend.title = element_text(size = 16),
        legend.text = element_text(size = 16),
        strip.text = element_text(size = 16))
```

Scatter plot

Code Figure

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



Simulate a more reasonable dataset

Multivariate normal distribution Code Figure

```r
mean_vec <- c(70, 70)
cov_mat <- matrix(c(25, 15, 15, 25), nrow = 2)
math_physics <- MASS::mvrnorm(n = n_students,
                              mu = mean_vec, Sigma = cov_mat)
math_physics <- round(math_physics)
math_physics[math_physics < 55] <- 55
math_physics[math_physics > 95] <- 95
grading_system[, c("math", "physics")] <- math_physics
```

Simulate a more reasonable dataset

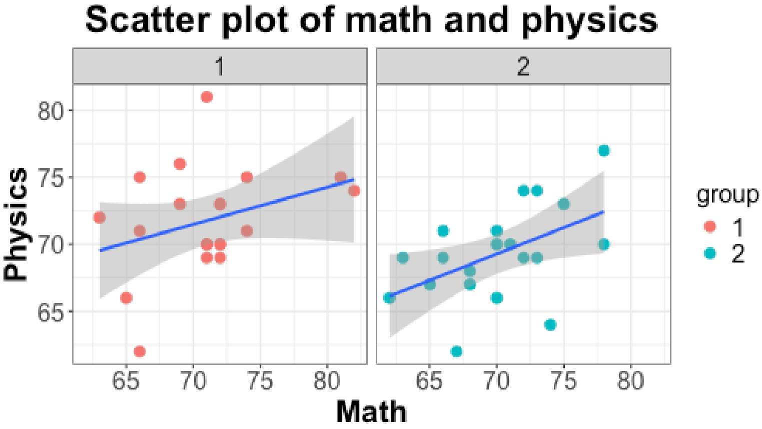
Multivariate normal distribution Code Figure

```
ggplot(grading_system[grading_system$group %in% c(1, 2), ], aes(x = math
  facet_grid(. ~ group) +
  geom_point(aes(color = group), size = 3) +
  geom_smooth(method = "lm") +
  labs(title = "Scatter plot of math and physics", x = "Math", y = "Phys
  coord_fixed(ratio = 1) +
  theme_bw() +
  theme(plot.title = element_text(face = "bold", size = 24, hjust = 0.5)
        axis.title = element_text(face = "bold", size = 20),
        axis.text = element_text(size = 16),
        axis.text.x = element_text(size = 16),
        axis.text.y = element_text(size = 16),
        legend.title = element_text(size = 16),
        legend.text = element_text(size = 16),
        strip.text = element_text(size = 16))
```

Simulate a more reasonable dataset

Multivariate normal distribution Code Figure

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



Thank you!

Slides created via Yihui Xie's R package **xaringan**.

Theme customized via Garrick Aden-Buie's R package **xaringanthemer**.

Tabbed panels created via Garrick Aden-Buie's R package **xaringanExtra**.

The chakra comes from remark.js, **knitr**, and R Markdown.