

# 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 <- 50; 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", "physics")])
knitr::kable(head(grading_system, n = 3), format = 'html')
```

id	group	gender	math	english	physics	average
1	1	F	69	80	76	75
2	2	M	86	86	68	80
3	3	M	85	59	63	69

# Some simple descriptive statistics

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

F M 23 27

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

math	english	physics
Min. :56.00	Min. :58.0	Min. :57.00
1st Qu.:63.25	1st Qu.:68.5	1st Qu.:62.25
Median :71.00	Median :77.0	Median :76.50
Mean :74.22	Mean :76.6	Mean :75.18
3rd Qu.:85.00	3rd Qu.:86.0	3rd Qu.:86.75
Max. :95.00	Max. :94.0	Max. :94.00

# Some simple descriptive statistics

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

	math	english	physics
math	162.17510	-15.930612	-15.836327
english	-15.93061	115.265306	8.277551
physics	-15.83633	8.277551	151.415918

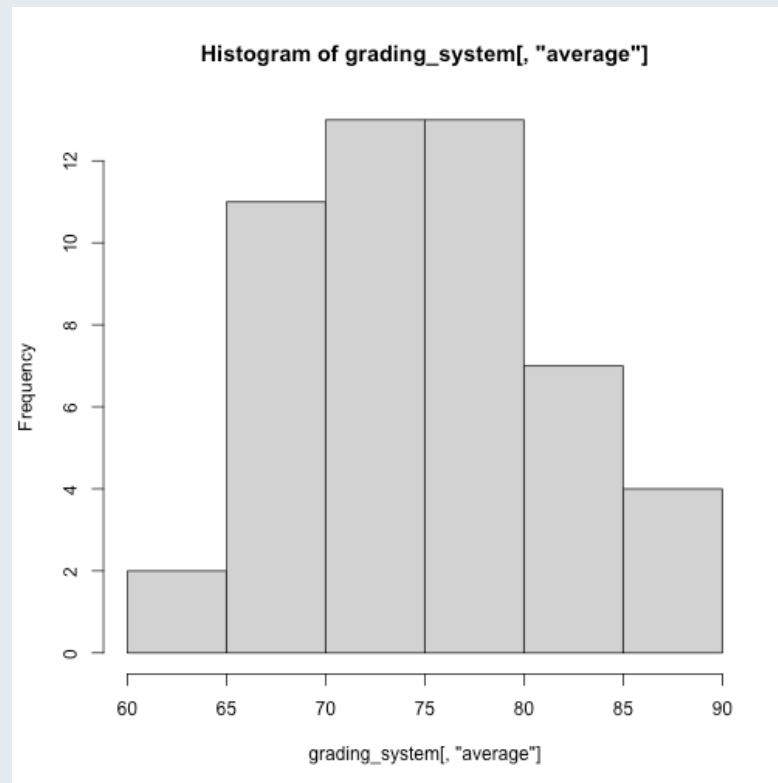
```
knitr::kable(cor(grading_system[, c("math", "english", "physics"))), for
```

	math	english	physics
math	1.0000000	-0.1165175	-0.1010594
english	-0.1165175	1.0000000	0.0626566
physics	-0.1010594	0.0626566	1.0000000

**How about the visualization?**

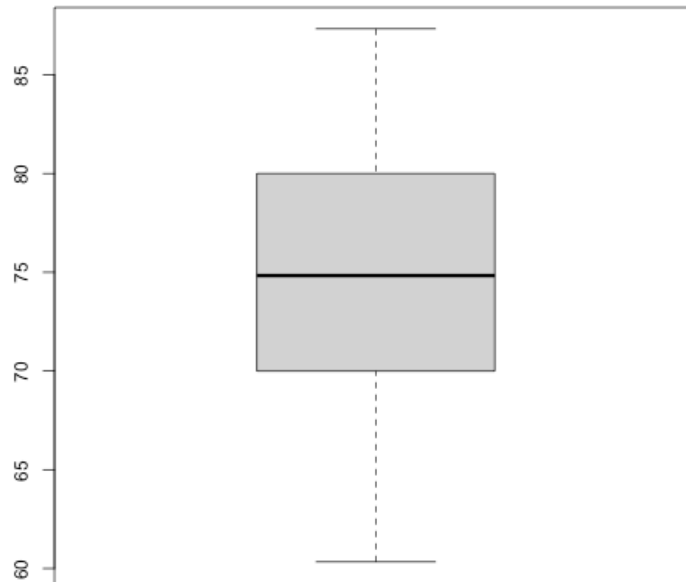
# Histogram

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



# Boxplot

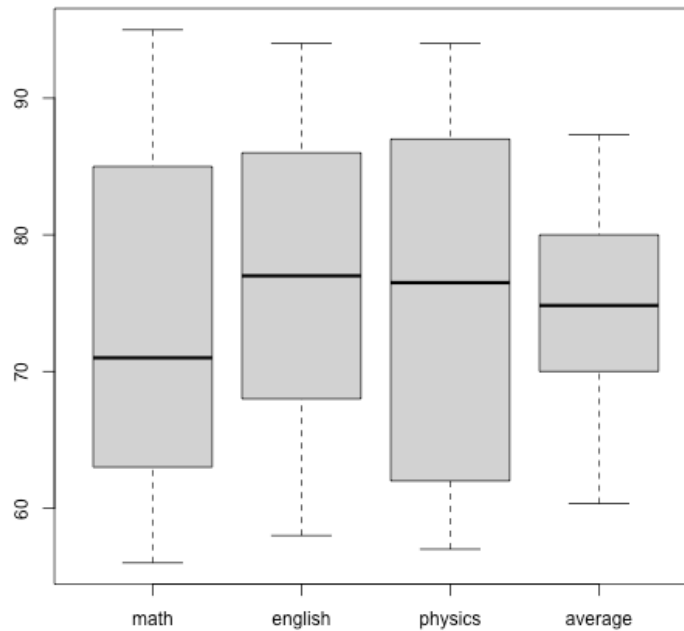
```
boxplot(grading_system[, "average"])
```





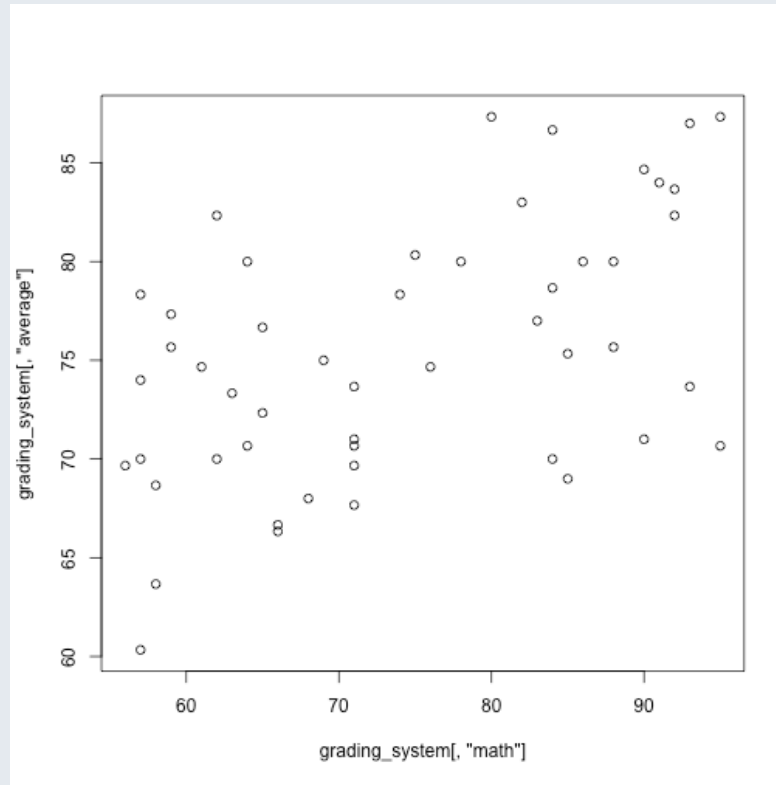
# Boxplot

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



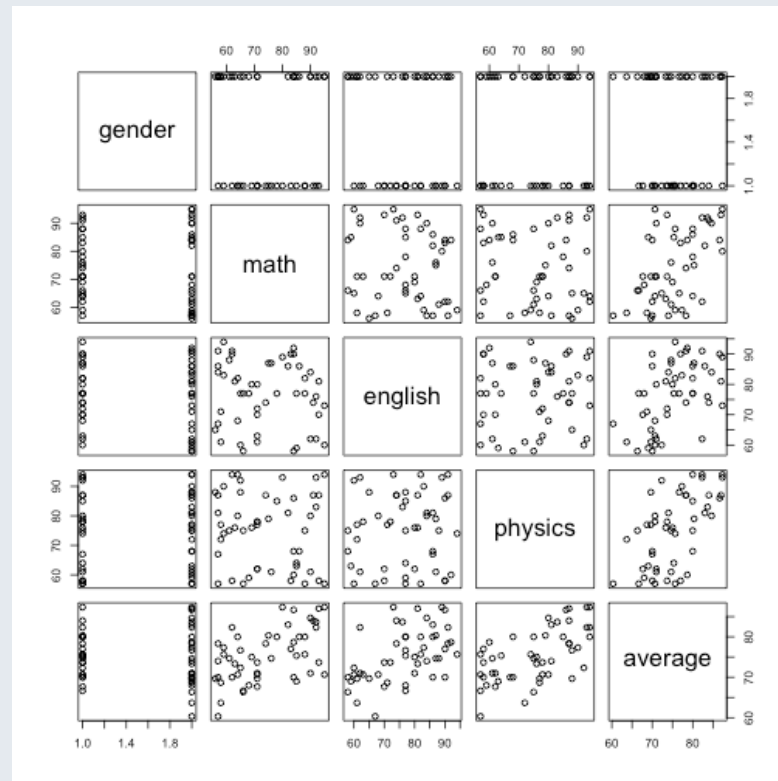
# Scatterplot

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



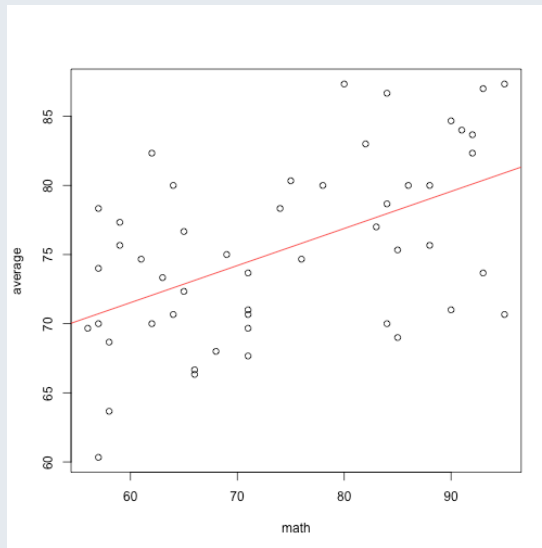
# Scatterplot

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



# Scatterplot

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



```
fit_lm$coefficients
```

(Intercept) math 55.4394167 0.2680398

# 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

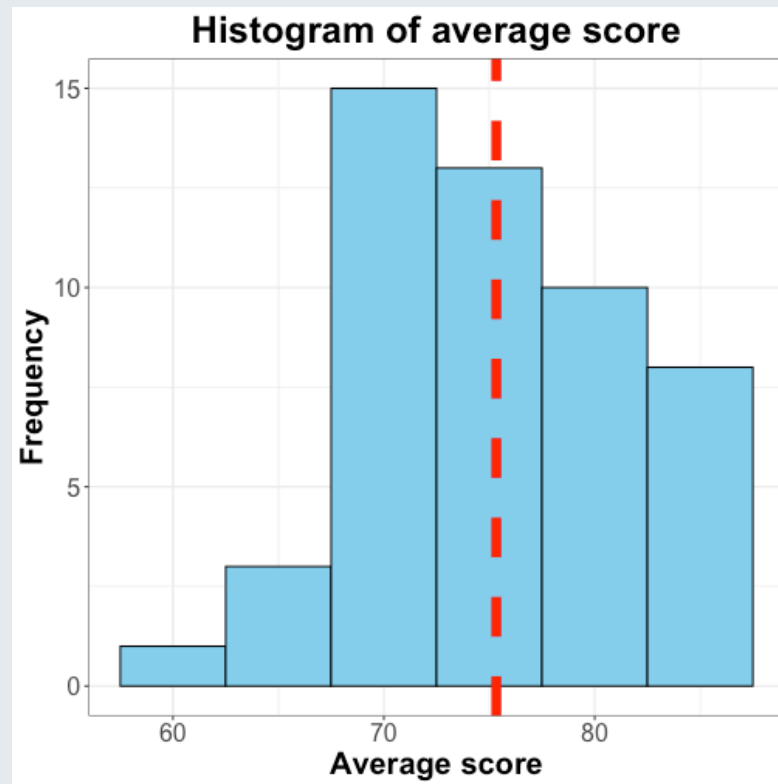
---

```
# 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 = "I
  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

```
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 = "Frequency") +  
  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))
```

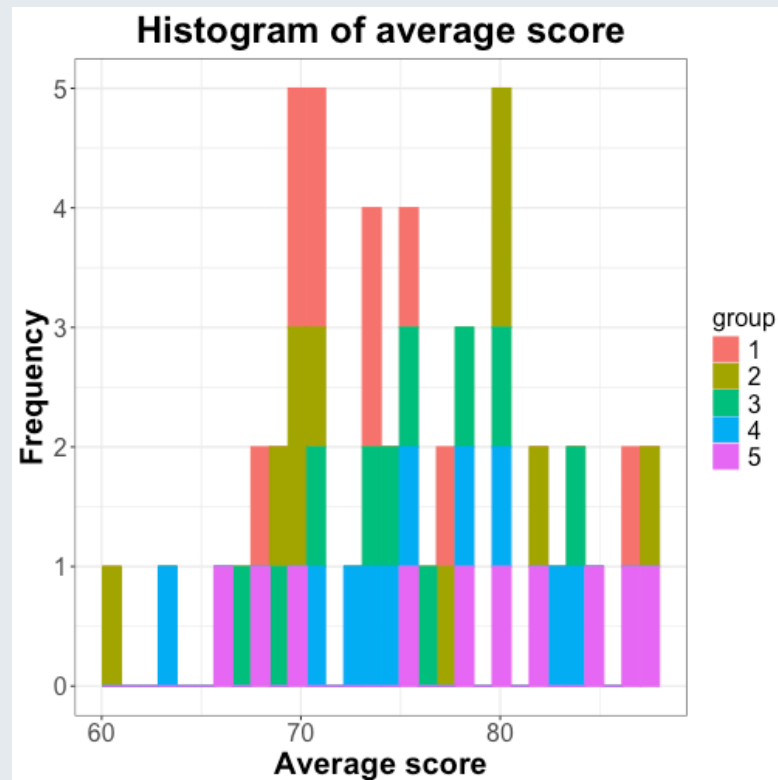
# Histogram in group

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

```
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"))
```

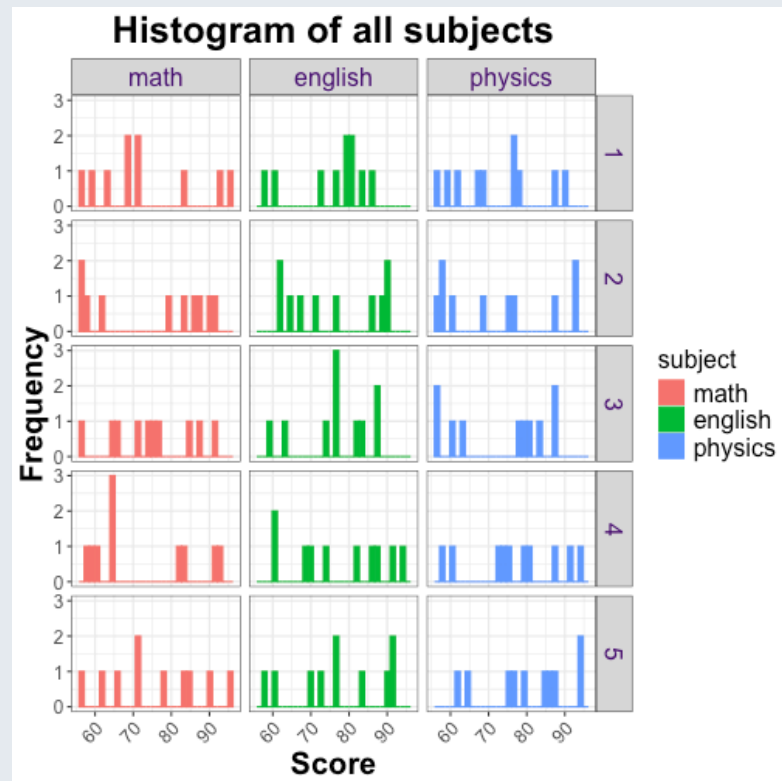
# Histogram with `facet_grid()`

`facet_grid()`

Code

Figure

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



# Boxplot

---

Code

Figure

---

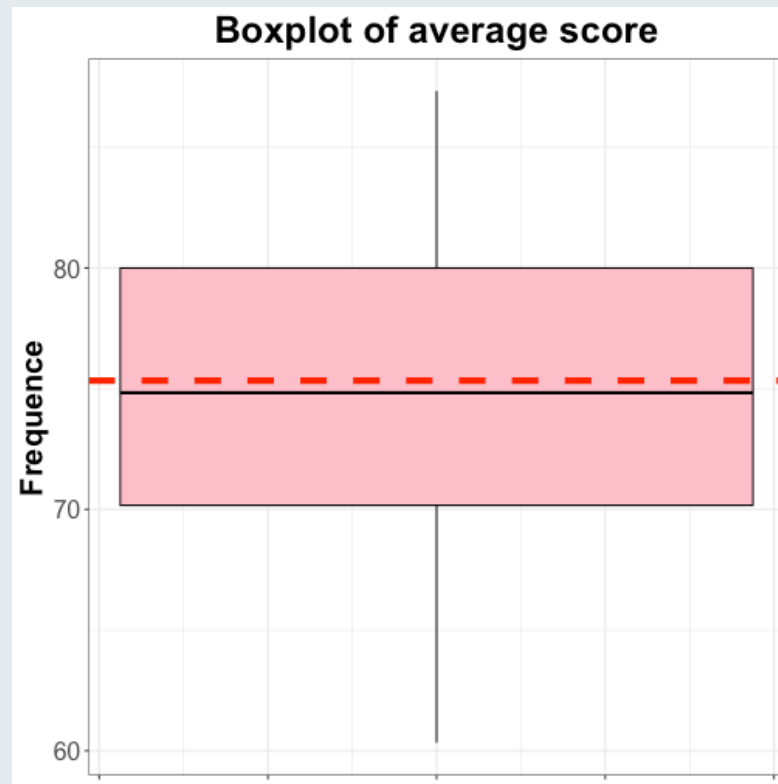
```
ggplot(grading_system, aes(y = average)) +  
  geom_boxplot(color = "black", fill = "pink") +  
  geom_hline(aes(yintercept = mean(average)), color = "red", linetype =  
  labs(title = "Boxplot of average score", y = "Frequency") +  
  theme_bw() +  
  theme(plot.title = element_text(face = "bold", 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))
```

---

# Boxplot

Code

Figure



# Boxplot in group

---

Code

Figure

---

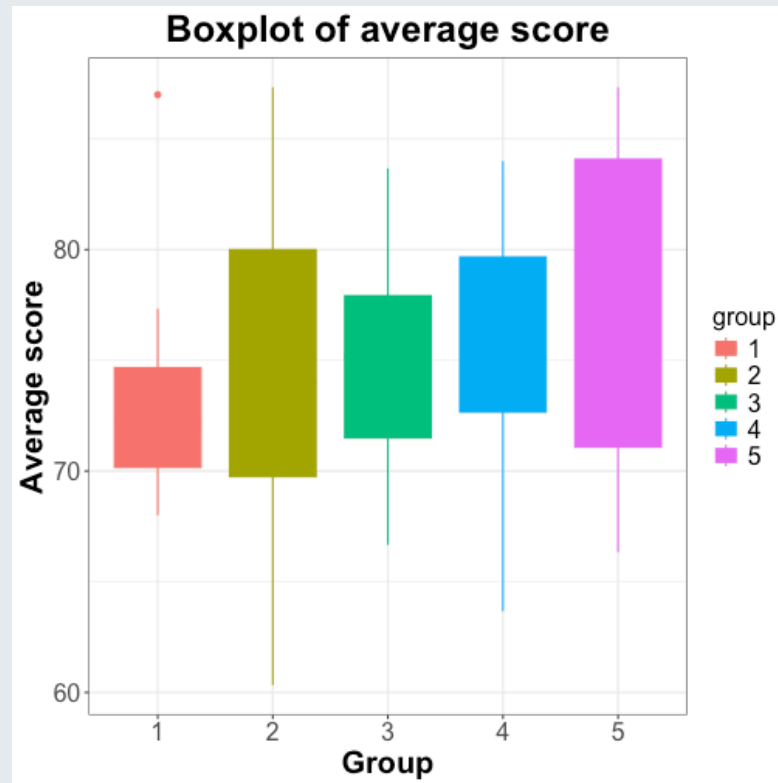
```
ggplot(grading_system, aes(x = group, y = average)) +  
  geom_boxplot(aes(fill = group, color = group)) +  
  labs(title = "Boxplot of average score", x = "Group", y = "Average score") +  
  theme_bw() +  
  theme(plot.title = element_text(face = "bold", size = 24, hjust = 0.5),  
        axis.title = 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))
```



# Boxplot in group

Code

Figure



# Boxplots with `facet_grid()`

---

Code

Figure

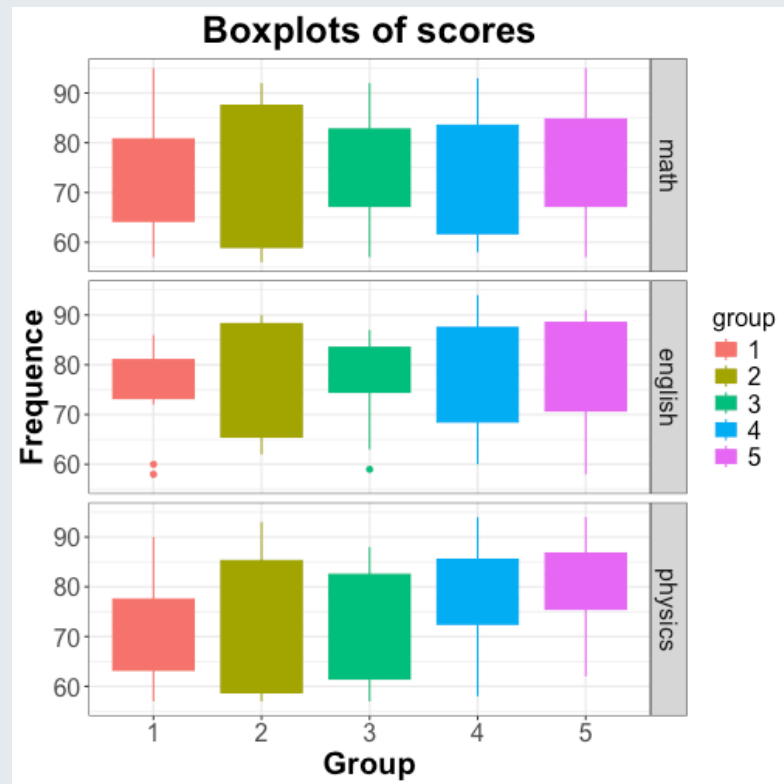
---

```
ggplot(grading_system_long, aes(x = group, y = score)) +  
  geom_boxplot(aes(fill = group, color = group)) +  
  labs(title = "Boxplots of scores", x = "Group", y = "Frequency") +  
  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))
```

# Boxplots with `facet_grid()`

Code

Figure



# Violin plot

---

Code

Figure

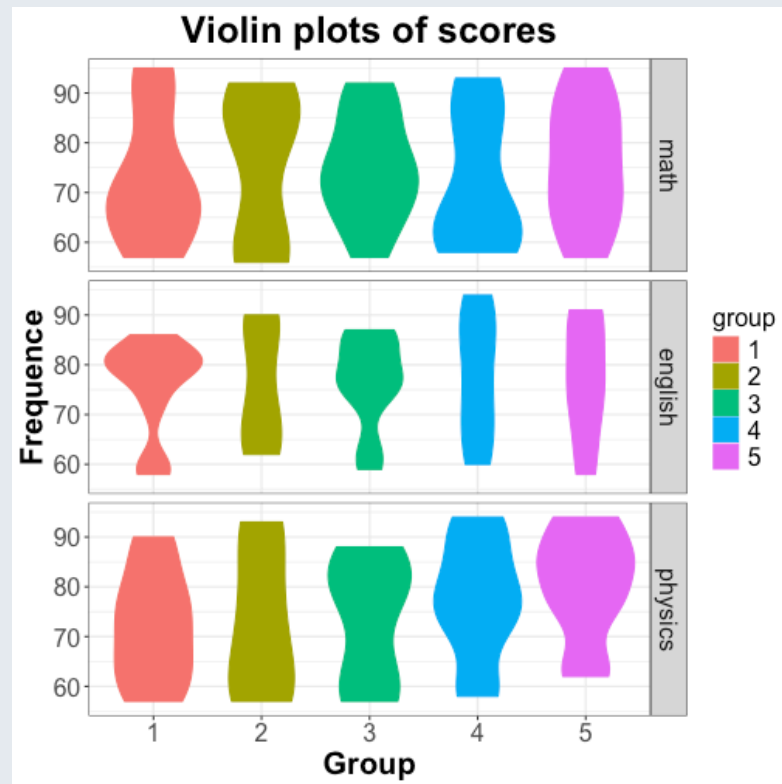
---

```
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 = "Frequency") +  
  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 boxplot inner

---

Code

Figure

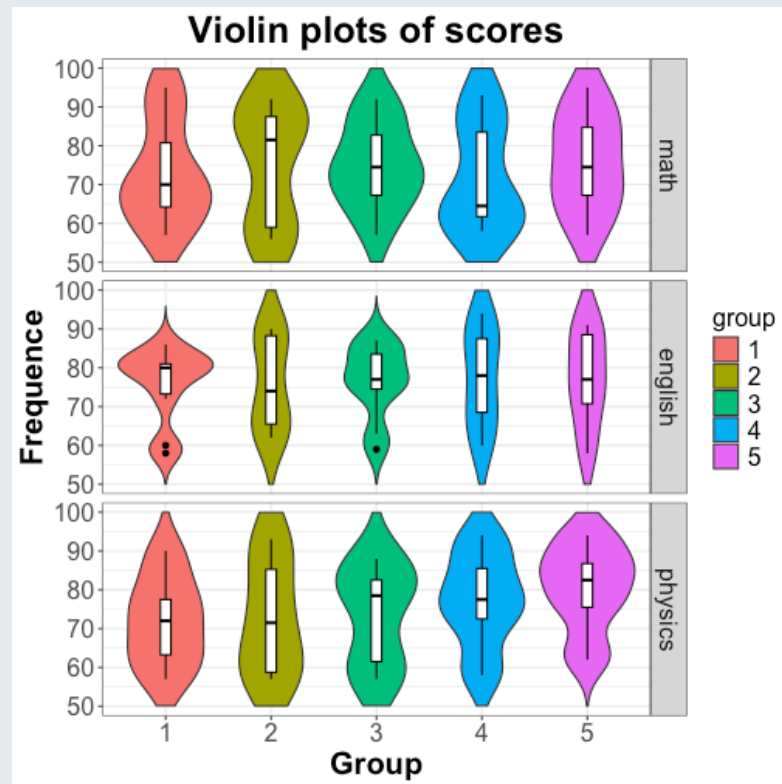
---

```
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 = "Frequency") +  
  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 boxplot inner

Code

Figure



# Scatter plot

---

Code

Figure

---

```
# Use only first two groups
ggplot(grading_system[grading_system$group %in% c(1, 2), ], aes(x = math, y = physics)) +
  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 = "Physics") +
  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))
```

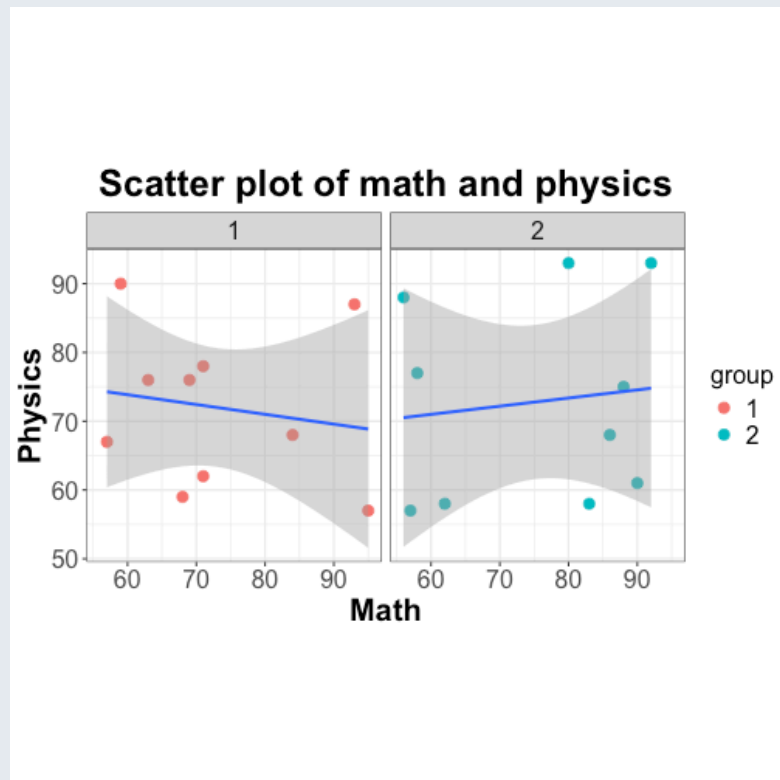


# Scatter plot

Code

Figure

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



# Generate a more reasonable dataset

---

Multivariate normal distribution

---

Code

Figure

---

```
mean_vec <- c(70, 70)
cov_mat <- matrix(c(25, 20, 20, 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
```

# Generate a more reasonable dataset

Multivariate normal distribution

Code

Figure

```
ggplot(grading_system[grading_system$group %in% c(1, 2), ], aes(x = math, y = physics)) +  
  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 = "Physics") +  
  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))
```

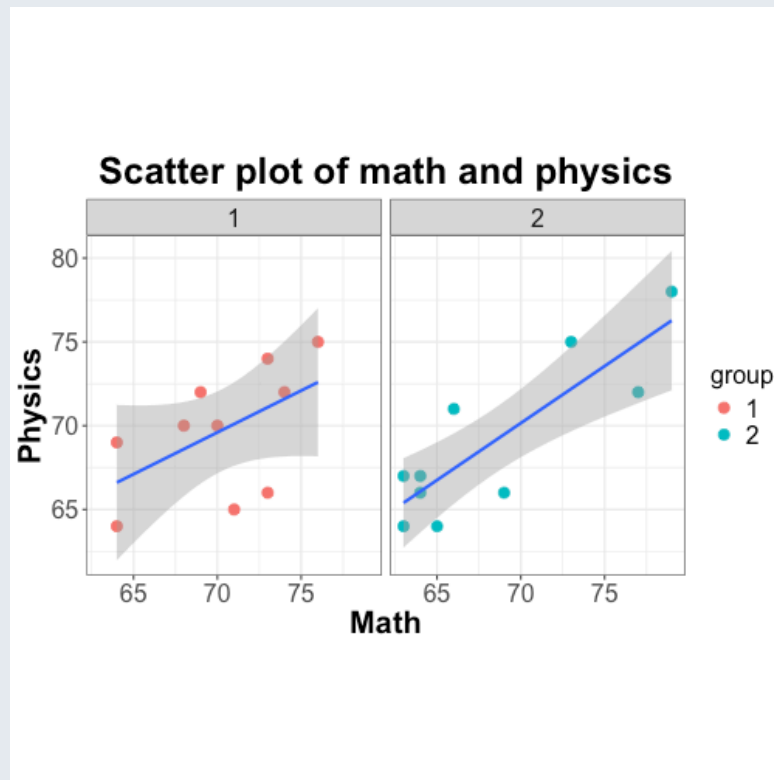
# Generate a more reasonable dataset

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 [xaringanthemmer](#).

Tabbed panels created via Garrick Aden-Buie's R package [xaringanExtra](#).

The chakra comes from [remark.js](#), [knitr](#), and [R Markdown](#).