Intro to Data Visualization and Statistics in R

Session #6

Genome Institute of Singapore

6th Nov 2019

Learning objectives for Session 6

Objectives:

A very gentle introduction to statistical modelling and testing.

Learn the following concepts

1. Univariate statistics: t-test, correlation, ANOVA.
2. Linear models

Some popular univariate statistics

Outcome	Independent variable	Example	Parametric test	Non-parametric test
Continuous	2 groups (paired)	Gene expression changes after intervention (same subjects)	one sample or paired t-test	Wilcoxon Signed-rank sum test
	2 groups (independent)	Gene expression ~ Gender	two sample t-test	Wilcoxon rank sum test / Mann Whitney U test
	> 2 groups	Gene expression ~ Blood group	ANOVA	Kruskal Wallis
	Continuous	Gene expression ~ Protein marker	Peason correlation	Spearman rho correlation

All of the above can be tested via linear regression model

Outcome	Independent variable	Example	Parametric test	
Discrete	2 groups (paired)	Gestational diabetes in pregnancy 1 vs pregnancy 2 (same subjectes)	McNemar's test	
	2 groups (independent)	Genotype ~ case/control status	Fisher's Exact test	
	> 2 groups	Genotype ~ Blood group	Chi-squared test	
	Continuous	Diabetes status ~ BMI	Logistic regression model	

All of the above can be tested via logistic regression model

We will look at two sample t-tests, ANOVA, pearson correlation and linear models today.

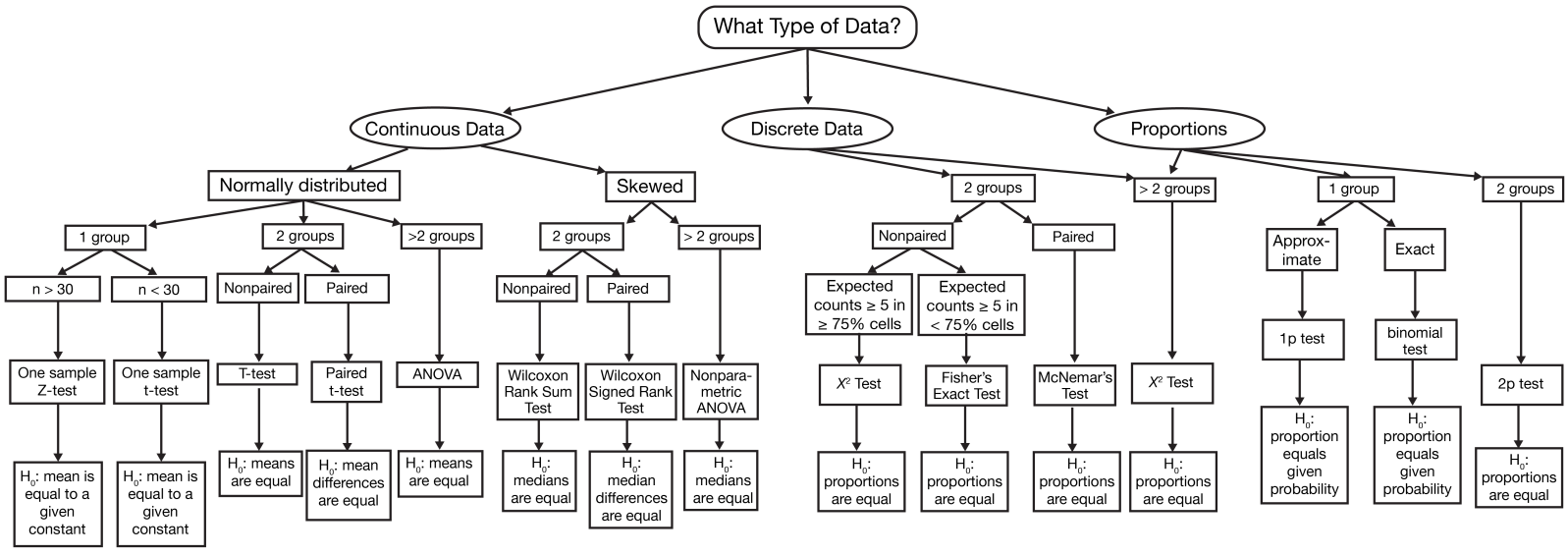
Other outcomes: proportions, survival data, ordinal variables etc.

Plenty of resources on how to choose statistical tests

For example:

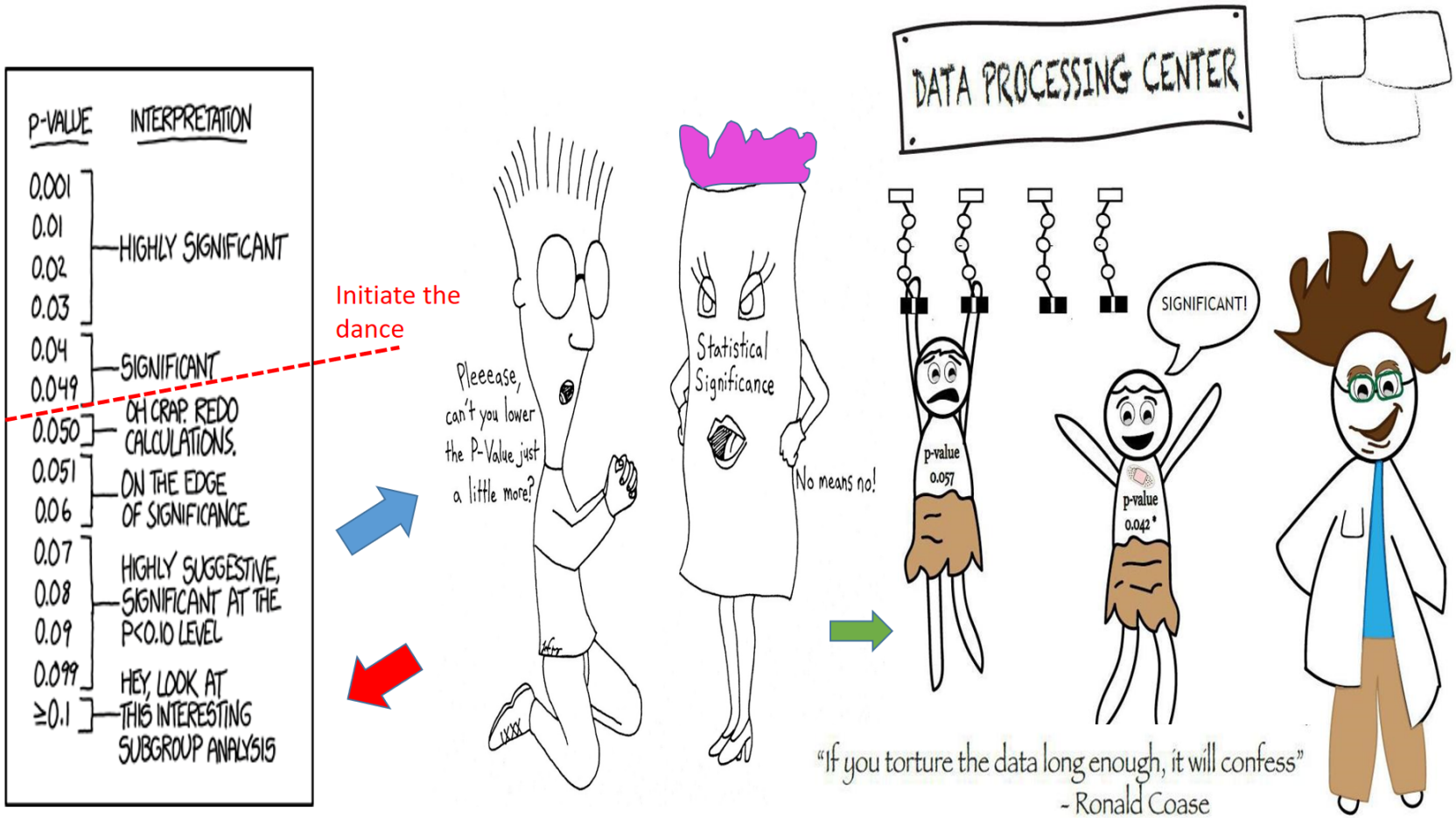
- https://onishlab.colostate.edu/wp-content/uploads/2019/07/hypothesis_testing.png
- https://statranalysis.net/2015/07/27/choosing-the-correct-statistical-test/
- https://data-flair.training/blogs/hypothesis-testing-in-r/
- Plus most of the drop down commercial softwares (Stata, Prism, etc)

Flow chart: which test statistic should you use?



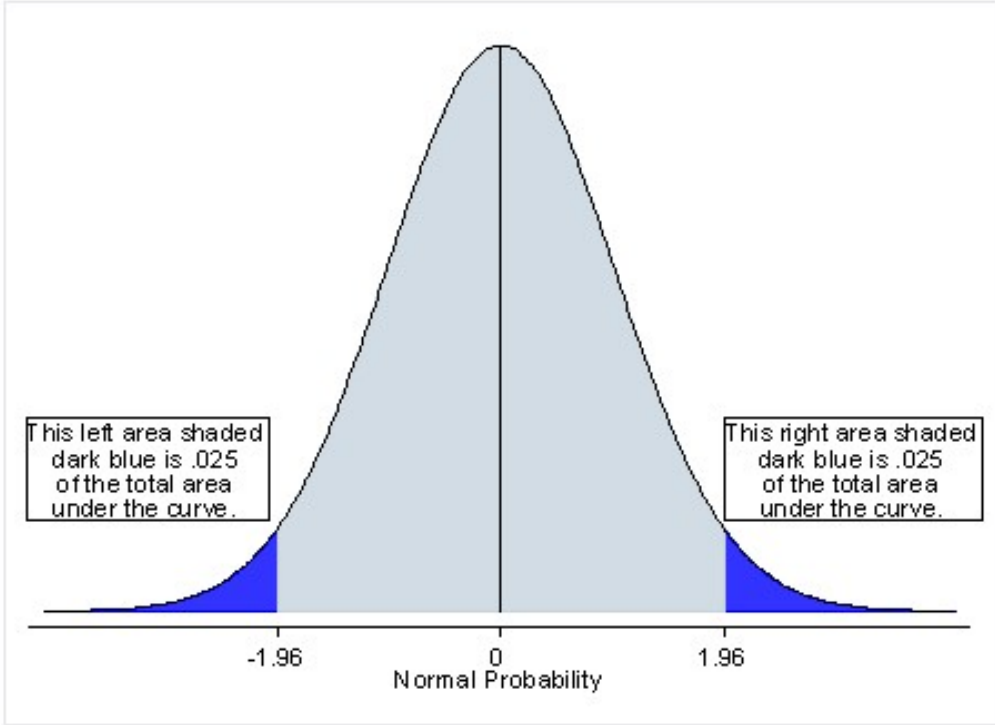
Source: https://onishlab.colostate.edu/wp-content/uploads/2019/07/which_test_flowchart.png

Dance of the p-values



P-value definition

P-value is the probability of obtaining a **test statistics** $\geq$ observed statistics assuming the null hypothesis is true.



1. The test statistics and distribution depends on the test.
2. Significance depends on one-tail (testing superiority only) or two-tail test (testing differences; depicted in picture above).

Test statistics for t-test

For Normal distributed outcome with the following assumptions:

- Equal sample sizes (n subjects in each group)
- Equal variance in both groups

the t-test statistics reduces to:

$$t_{obs} = \frac{m_1 - m_2}{\sqrt{v_1 + v_2}} \times \sqrt{n}$$

where m_i and v_i are the mean and variance of the i^{th} group.

Test statistics for t-test

For Normal distributed outcome with the following assumptions:

- Equal sample sizes (n subjects in each group)
- Equal variance in both groups

the t-test statistics reduces to:

$$t_{obs} = \frac{m_1 - m_2}{\sqrt{v_1 + v_2}} \times \sqrt{n}$$

where m_i and v_i are the mean and variance of the i^{th} group.

Or in English:

$$t_{obs} = \frac{\text{Difference in group means}}{\sqrt{\text{total variation}}} \times \sqrt{\text{sample size per group}}$$

Larger fold change, larger sample size and smaller variation corresponds to larger t-statistics (i.e. more significant p-value).

Aside: Power calculation for t-test

Re-arrange the equation in previous slide gives us

$$\text{sample size per group} = \frac{\text{total variation}}{(\text{Difference in group means})^2} \times t_{obs}^2$$

Therefore, you will need larger sample size if

Aside: Power calculation for t-test

Re-arrange the equation in previous slide gives us

$$\text{sample size per group} = \frac{\text{total variation}}{(\text{Difference in group means})^2} \times t_{obs}^2$$

Therefore, you will need larger sample size if

- your data is likely to be noisy ($\uparrow$ total variation)
- the biological difference is harder to detect ($\downarrow$ difference)
- the statistical significance you wish to report is smaller ($\uparrow t_{obs}$ which means $\downarrow p$)

Aside: Power calculation for t-test

Re-arrange the equation in previous slide gives us

$$\text{sample size per group} = \frac{\text{total variation}}{(\text{Difference in group means})^2} \times t_{obs}^2$$

Therefore, you will need larger sample size if

- your data is likely to be noisy ($\uparrow$ total variation)
- the biological difference is harder to detect ($\downarrow$ difference)
- the statistical significance you wish to report is smaller ($\uparrow t_{obs}$ which means $\downarrow p$)

1. Many softwares available to calculate sample size requirements. E.g. G*Power, online calculators.
2. Avoid retrospective power calculations.

Height Weight dataset

Height weight | Preparation

1. https://github.com/adairama/R_workshop_GIS/
2. Click on Session 6
3. Download or copy `height_weight_Session6_partial.R`
4. Change path in `setwd()` and `read_excel()` lines.
5. Run until the `skim()` command

We will use the weight as the outcome here. Your weight should range between 29.3 - 122.5kg.

Your task 1: Visualize the weight distribution by gender Write your codes below the **"YOUR TASK 1"** section in the script.

Height weight | Summary of weight

Here is one way to calculate the mean and standard deviation (sd) for all 10,000 participants.

```
hw %>%
  summarise(count = n(),
            mean_weight = mean(Weight),
            sd_weight = sd(Weight) ) %>%
  kable(digits=1)
```

count	mean_weight	sd_weight
10000	73.2	14.6

Your task 2: Calculate the mean and sd for males and females separately.

Height weight | Summary of weight by gender

You can use `filter(Gender=="Male")` to get the summaries and then repeat with `filter(Gender=="Female")`. However, `group_by` is much more concise:

```
hw %>%
  group_by(Gender) %>%
  summarise(count = n(),
            mean_weight = mean(Weight),
            sd_weight   = sd(Weight)) %>%
  kable(digits=1)
```

Gender	count	mean_weight	sd_weight
Female	5000	61.6	8.6
Male	5000	84.8	9.0

```
61.6 - 84.8  ## Weight difference between female and male
## [1] -23.2
```

Height weight | t-test statistics calculation by hand

Gender	count	mean_weight	sd_weight
Female	5000	61.6	8.6
Male	5000	84.8	9.0

Using the equation earlier:

$$t_{obs} = \frac{m_1 - m_2}{\sqrt{v_1 + v_2}} \times \sqrt{n}$$

we get t_{obs} and the p-value as:

```
stat <- ((61.6 - 84.8)/sqrt(8.63^2 + 8.97^2)) * sqrt(5000)
stat
## [1] -131.7936

2*pnorm(-abs(stat)) # p-value < 1e-325 (gets rounded to 0)
## [1] 0
```

Height weight | t-test function (base R approach)

```
t.test(Weight ~ Gender, data=hw)
#   Welch Two Sample t-test
#
# data:  Weight by Gender
# t = -131.82, df = 9982.8, p-value < 2.2e-16
# alternative hypothesis:
#   true difference in means is not equal to 0
#
# 95 percent confidence interval:
#  -23.54708 -22.85704
#
# sample estimates:
# mean in group Female   mean in group Male
#             61.61434             84.81640
```

It is a little hard to extract the test statistics, p-values etc from the model above. Plus the weight difference is not available.

Height weight | t-test function (tidyverse approach)

We can use `tidy()` from the `broom` package:

```
hw %>%
  t.test(Weight ~ Gender, data=.) %>%      ## see notes
  tidy() %>%                               ## the main hero
  select(estimate1, estimate2, diff=estimate,
         statistic, p.value)
## # A tibble: 1 x 5
##   estimate1 estimate2  diff statistic p.value
##       <dbl>     <dbl> <dbl>     <dbl>   <dbl>
## 1      61.6      84.8 -23.2     -132.       0
```

1. If `data` is not the first argument in a function, you need to include "data=." into the command.
2. `tidy()` works on a whole host of statistical outputs to get structured outputs. See `help(tidy)`.

Height weight | Generalize to linear models

You can generalize the t-test statistics to a linear model.

The estimate for Gender uses females as reference. Thus the estimate (23.2kg), statistics (132) and p-value (0) are the same.

```
hw %>%
  lm(Weight ~ Gender, data=.) %>%
  tidy()
## # A tibble: 2 x 5
##   term         estimate std.error statistic p.value
##   <chr>           <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)     61.6      0.124      495.       0
## 2 GenderMale      23.2      0.176      132.       0
```

Expected(Weight | Female) = 61.6kg
Expected(Weight | Male) = 61.6 + 23.2 = 84.8kg

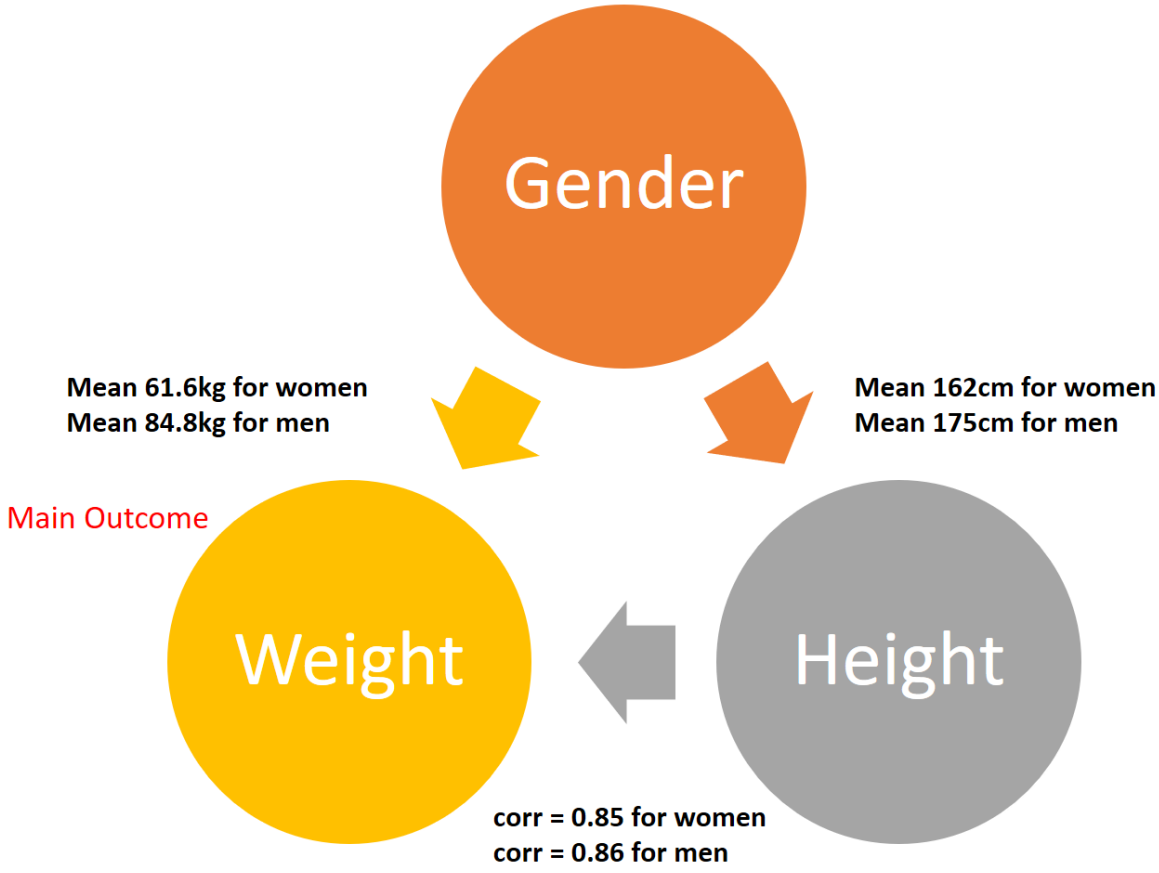
Height weight | Can you publish it?

Can you simply report that the weight differences between a man and woman is 23.2kg?

Height weight | Can you publish it?

Can you simply report that the weight differences between a man and woman is 23.2kg?

No, because height also have a strong contribution to weight.



Other unaccounted factors (e.g. age, ethnicity, bone density) may have further influence on weight.

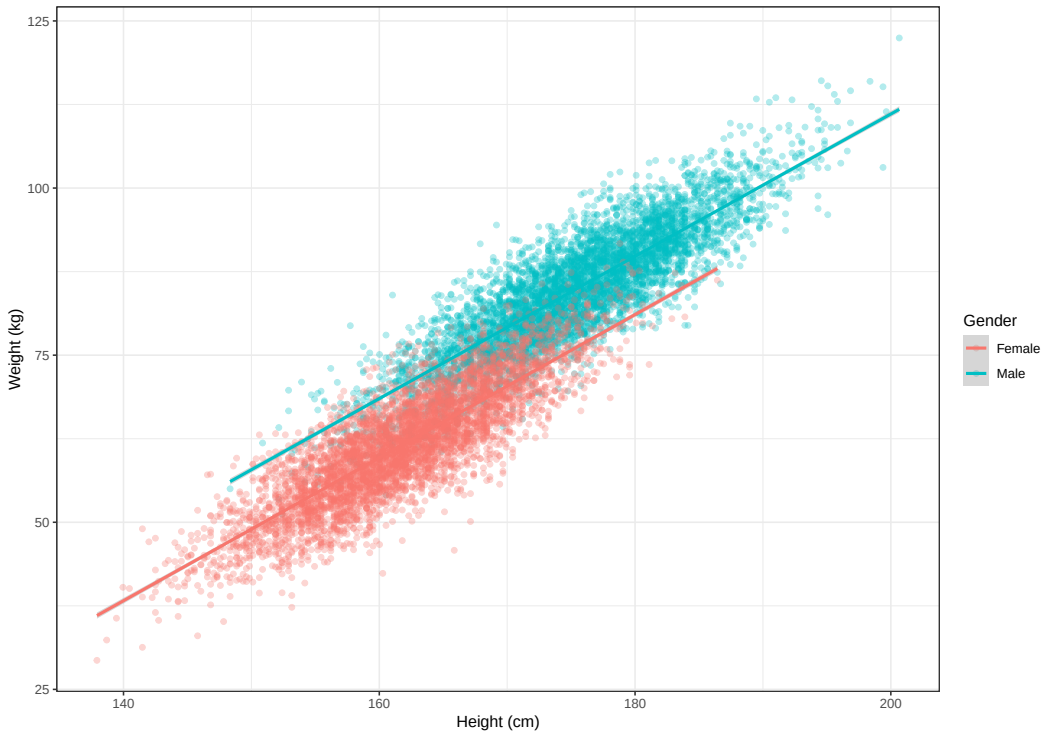
Height weight | Scatterplot

Your task 3: Plot weight vs height and color code by gender.

Height weight | Scatterplot

Your task 3: Plot weight vs height and color code by gender.

```
ggplot(hw, aes(x=Height, y=Weight, col=Gender)) +
  geom_point(alpha=0.3) +
  geom_smooth(method="lm") +
  labs(x="Height (cm)", y="Weight (kg)") +
  theme_bw()
```



Height weight | Correlation

Code using base R grammar:

```
cor( hw$Height, hw$Weight )
## [1] 0.9247714
```

Code for using tidyverse grammar which is more flexible:

```
hw %>%
  summarise(cor(Height, Weight))
## # A tibble: 1 x 1
##   `cor(Height, Weight)`
##                   <dbl>
## 1                 0.925
```

Your task 4: Calculate correlation by gender.

Height weight | Correlation by Gender

Your task 4: Calculate correlation by gender.

Again, `group_by()` is more concise that `filter()` here:

```
hw %>%
  group_by(Gender) %>%
  summarise(cor(Height, Weight))
## # A tibble: 2 x 2
##   Gender `cor(Height, Weight)`
##   <chr>                  <dbl>
## 1 Female                 0.850
## 2 Male                   0.863
```

Height weight | Weight diff by Gender within a height bin

Your task 5: Calculate the number of males and females who are 167cm to 171cm tall and their respective group average weights.

Height weight | Weight diff by Gender within a height bin

Your task 5: Calculate the number of males and females who are 167cm to 171cm tall and their respective group average weights.

```
hw %>%
  filter(Height > 167, Height < 171) %>%
  group_by(Gender) %>%
  summarise(count = n(),
            mean_weight = mean(Weight)) %>%
  kable(digits=1)
```

Gender	count	mean_weight
Female	693	69.1
Male	724	78.4

```
78.4 - 69.1    ## Weight diff in this bin
## [1] 9.3
```

Height weight | Weight diff within height quintiles (part 1)

```
q5 <- quantile(hw$Height, seq(0, 1, by=0.2))
q5
##      0%     20%     40%     60%     80%    100%
## 137.922 159.766 165.608 171.196 177.292 200.660

hw <- hw %>%
  mutate(Height_bin = cut(Height, q5, include.lowest=T))

hw %>% tabyl(Height_bin)  # roughly equal sized bins
##  Height_bin    n percent
##    [138,160] 2057  0.2057
##    (160,166] 2006  0.2006
##    (166,171] 1953  0.1953
##    (171,177] 2007  0.2007
##    (177,201] 1977  0.1977
```

Height weight | Weight diff within height quintiles (part 2)

Let's check the distribution of the gender by height quintiles.

```
hw %>%
  tabyl(Height_bin, Gender) %>%
  adorn_percentages("row") %>%
  adorn_pct_formatting(digits=1) %>%
  adorn_ns("front") %>%
  kable()
```

Height_bin	Female	Male
[138,160]	1958 (95.2%)	99 (4.8%)
(160,166]	1626 (81.1%)	380 (18.9%)
(166,171]	1010 (51.7%)	943 (48.3%)
(171,177]	353 (17.6%)	1654 (82.4%)
(177,201]	53 (2.7%)	1924 (97.3%)

Height weight | Weight diff within height quintiles (part 3)

Calculate the weight difference and run *t*-test in each quintile.

```
hw %>%
  group_by(Height_bin) %>%
  do(tidy( t.test(Weight ~ Gender, data=.))) %>%
  select(Height_bin, mean_F=estimate1, mean_M=estimate2,
         weight_diff=estimate, statistic, p.value) %>%
  kable(digits=1)
```

Height_bin	mean_F	mean_M	weight_diff	statistic	p.value
[138,160]	54.4	65.4	-11.0	-20.5	0
(160,166]	62.7	72.4	-9.7	-35.6	0
(166,171]	68.3	78.1	-9.8	-43.5	0
(171,177]	74.4	83.8	-9.3	-33.8	0
(177,201]	80.8	92.5	-11.6	-15.7	0

The `do()` function is required to fit models per group.

Height weight | Linear model

Your task 6: Add Gender into the linear model and summarize it.

Height weight | Linear model

Your task 6: Add Gender into the linear model and summarize it.

```
lm( Weight ~ Height + Gender, data=hw ) %>%
  tidy() %>%
  kable(digits=2)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-111.06	1.04	-106.55	0
Height	1.07	0.01	165.98	0
GenderMale	8.79	0.13	69.92	0

We expect males to be 8.79kg heavier than females of same height.

Height weight | Linear model

Your task 6: Add Gender into the linear model and summarize it.

```
lm( Weight ~ Height + Gender, data=hw ) %>%
  tidy() %>%
  kable(digits=2)
```

term	estimate	std.error	statistic	p.value
(Intercept)	-111.06	1.04	-106.55	0
Height	1.07	0.01	165.98	0
GenderMale	8.79	0.13	69.92	0

We expect males to be 8.79kg heavier than females of same height.

Expected(Weight | Female) = -111 + 1.07 x Height_cm
Expected(Weight | Male) = -111 + 1.07 x Height_cm + 8.79

Note: Height is between 137 - 200cm in this population.

Height weight | Predictions using a linear model

Your task 7: What is the expected weight for a 150cm tall man?

Height weight | Predictions using a linear model

Your task 7: What is the expected weight for a 150cm tall man?

We can do this manually:

```
-111 + 1.07*150 + 8.79
## [1] 58.29
```

or using the `predict()` method in R

```
fit    <- lm( Weight ~ Height + Gender, data=hw )
testdf <- data.frame(Gender="Male", Height=150)
testdf
##   Gender Height
## 1   Male    150

predict(fit, testdf) ## not exact due to rounding above
##        1
## 57.78882
```

Height weight | Predictions using a linear model

You can also make predictions on multiple subjects in parallel:

```
x <- seq(130, 200, by=10)
x
## [1] 130 140 150 160 170 180 190 200

testM <- data.frame(Gender="Male", Height=x)
testM$exp_weight_male <- predict(fit, testM)
testM <- testM %>% select(-Gender)

testF <- data.frame(Gender="Female", Height=x)
testF$exp_weight_female <- predict(fit, testF)
testF <- testF %>% select(-Gender)
```

Height weight | Predictions using a linear model

Combining the outputs helps generate the following reference table:

```
full_join(testM, testF) %>%
  kable(digits=1)
```

Height	exp_weight_male	exp_weight_female
130	36.4	27.7
140	47.1	38.3
150	57.8	49.0
160	68.5	59.7
170	79.1	70.3
180	89.8	81.0
190	100.5	91.7
200	111.1	102.4

```
rm(x, testM, testF, fit, testdf, q5)
```

Iris dataset

Iris | Preparation & correlation

Load the built-in dataset with `data(iris)`

Your task 8: Calculate the correlation between Sepal Length and Sepal width for the whole dataset (i.e. ignoring Species info) and within each Species. Can you explain what you are finding?

Iris | Preparation & correlation

Load the built-in dataset with `data(iris)`

Your task 8: Calculate the correlation between Sepal Length and Sepal width for the whole dataset (i.e. ignoring Species info) and within each Species. Can you explain what you are finding?

```
iris %>%
  summarise(cor(Sepal.Length, Sepal.Width))
#   cor(Sepal.Length, Sepal.Width)
# 1                      -0.1175698

iris %>%
  group_by(Species) %>%
  summarise(cor(Sepal.Length, Sepal.Width))
#   Species    `cor(Sepal.Length, Sepal.Width)`
# 1 setosa                                0.743
# 2 versicolor                            0.526
# 3 virginica                             0.457
```

Iris | Scatterplot (part 1)

Your task 9: Plot the Sepal Length vs Sepal Width with and without color for species. In each plot, add `geom_smooth(method="lm")` in.

Iris | Scatterplot (part 1)

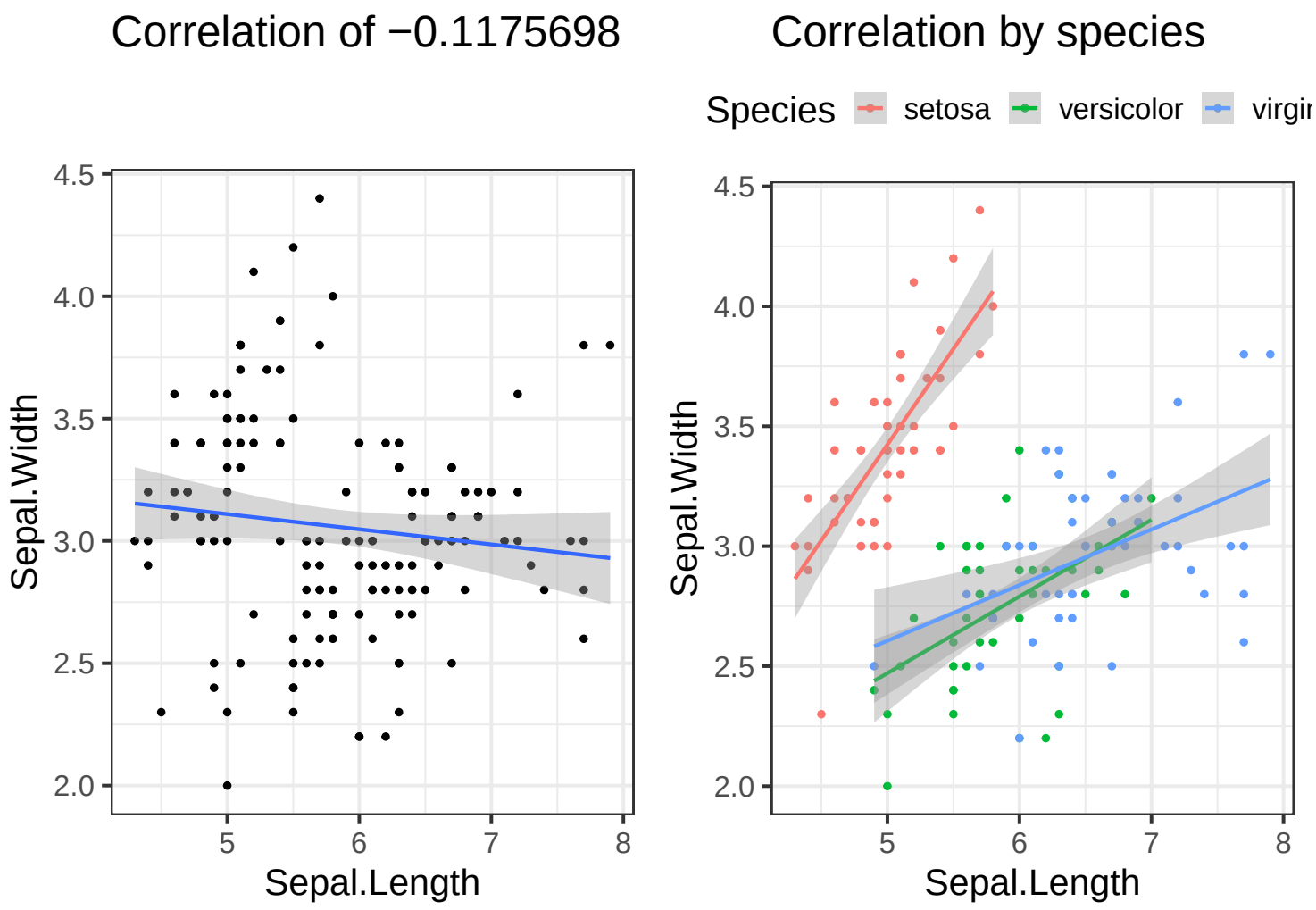
Your task 9: Plot the Sepal Length vs Sepal Width with and without color for species. In each plot, add `geom_smooth(method="lm")` in.

```
g1 <- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width)) +
  geom_point() +
  geom_smooth(method="lm") +
  ggtitle("Correlation of -0.1175698\n\n") +
  theme_bw(base_size=20)

g2 <- ggplot(iris, aes(x=Sepal.Length, y=Sepal.Width,
                       col=Species)) +
  geom_point() +
  geom_smooth(method="lm") +
  ggtitle("Correlation by species") +
  theme_bw(base_size=20) + theme(legend.position="top")
```

Iris | Scatterplot (part 2)

```
gridExtra::grid.arrange(g1, g2, nrow=1)
```



Iris | Linear model and ANOVA

Linear model and ANOVA are related. Linear models gives you the estimate of each level of the dependant variable. For example:

```
lm(Sepal.Length ~ Species, data=iris) %>% tidy()
#   term              estimate std.error   p.value
# 1 (Intercept)           5.01    0.0728 1.13e-113
# 2 Speciesversicolor     0.93    0.103  8.77e- 16
# 3 Speciesvirginica      1.58    0.103  2.21e- 32
```

ANOVA gives you the contribution of each dependent variable to the model. If a dependent variables is significant, you follow up with a post-hoc *t*-test to identify which levels are different.

```
aov(Sepal.Length ~ Species, data=iris) %>% tidy()
#   term          df sumsq meansq   p.value
# 1 Species        2  63.2 31.6    1.67e-31
# 2 Residuals    147  39.0  0.265        NA
```

Thank you