* [Question 1](https://elearn.waikato.ac.nz/pluginfile.php/3035398/mod_resource/content/1/Assn2-Ans.html" \l "question-1)
* [Question 2](https://elearn.waikato.ac.nz/pluginfile.php/3035398/mod_resource/content/1/Assn2-Ans.html#question-2)
* [Question 3](https://elearn.waikato.ac.nz/pluginfile.php/3035398/mod_resource/content/1/Assn2-Ans.html#question-3)
* [Question 4](https://elearn.waikato.ac.nz/pluginfile.php/3035398/mod_resource/content/1/Assn2-Ans.html#question-4)
* [Question 5](https://elearn.waikato.ac.nz/pluginfile.php/3035398/mod_resource/content/1/Assn2-Ans.html#question-5)

**Assignment 2**

DATAX201

**Question 1**

**Part (a)**  
We would expect a spreadsheet with information about individual observations spread across columns. This primarily violates the tidy data principle that variables are denoted by columns.

**Part (b)**  
We would expect a spreadsheet where variables about individual observations are spread across rows. This primarily violates the tidy data principle that observations are denoted by rows.

**Part (c)**  
When the observations are denoted by the columns of a spreadsheet and the variables are denoted by the rows of a spreadsheet. This violates the tidy data principles that observations are denoted by rows and variables are denoted by columns.

**Question 2**

library(dplyr)

library(ggplot2)

library(emmeans)

# Read in the data

paspalum.df <- read.table("data/paspalum.txt", header = TRUE)

# Fit a one-way analysis of variance

paspalum.fit <- lm(weight ~ temp, data = paspalum.df)

# Conduct a F-test

anova(paspalum.fit)

Analysis of Variance Table

Response: weight

Df Sum Sq Mean Sq F value Pr(>F)

temp 2 708.68 354.34 11.223 0.0001913 \*\*\*

Residuals 33 1041.90 31.57

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Use the emmeans() command to calculate the sample means with the appropriate 95% confidence intervals

paspalum.emm <- paspalum.fit %>%

emmeans( ~ temp) %>%

as\_tibble() %>%

# Let us use the x-axis label to communicate the units of temperature

mutate(temp = c("18", "22", "26"))

# Produce a plot of the sample means, whose error bars are the 95% confidence intervals

ggplot(paspalum.emm, aes(x = temp, y = emmean, ymin = lower.CL, ymax = upper.CL)) +

geom\_point(size = 3) +

geom\_errorbar(width = 0.25, linewidth = 1) +

theme\_bw() +

theme(text = element\_text(size = 12)) +

labs(x = "Temperature (°C)", y = "Average weight (grams)",

title = "Average weight of paspalum by glasshouse temperatures")

**Question 3**

**Part (a)**

library(dplyr)

library(tidyr)

library(readxl)

# The data as-is is "wide"...

lights.df <- read\_excel("data/lightFixtures.xlsx") %>%

# So, pivot it from "wide" to "long" to keep it consistent with tidy data principles

pivot\_longer(cols = starts\_with("Unit"), names\_to = "Within.Sample", values\_to = "Weight")

**Part (b)**

library(ggplot2)

# Create a histogram of the light fixtures

ggplot(lights.df, aes(x = Weight)) +

geom\_histogram(bins = 14, colour = "black", fill = "goldenrod") +

theme\_bw() +

labs(x = "Weight (grams)", y = "Count", title = "Distribution of light fixture weights")

**Part (c)**

# Let us print the sample mean rather than assume...

lights.df %>%

summarise(avg.weight = mean(Weight))

# A tibble: 1 × 1

avg.weight

<dbl>

1 169.

Noting that the sample mean was 169 grams, it seems reasonable to state that the light fixture weights are approximately symmetrically distributed about the sample mean weight. So we should trust the interpretation of any statistical inference made using this data.

**Part (d)**

t.test(Weight ~ 1, data = lights.df)

One Sample t-test

data: Weight

t = 245.81, df = 219, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

167.6675 170.3779

sample estimates:

mean of x

169.0227

**Part (e)**  
With 95% confidence, we estimate that the true mean weight of the light fixtures is somewhere between 167.7 and 170.4 grams.

**Question 4**

**Part (a)**

library(foreign)

birth.df <- read.spss("data/babies.sav", to.data.frame = TRUE)

*Alternatively*

library(haven)

birth.df <- read\_spss("data/babies.sav")

**Part (b)**

library(dplyr)

birth.tidied.df <- birth.df %>%

mutate(

# `gestation` is a numeric variable

gestation = if\_else(gestation == 999, NA, gestation),

# `parity` is a categorical variable

parity = if\_else(parity == 1, "First born", "Not first born"),

# `age` is a numeric variable

age = if\_else(age == 99, NA, age),

# `height` is a numeric variable

height = if\_else(height == 99, NA, height),

# `weight` is a numeric variable

weight = if\_else(weight == 999, NA, weight),

# `smoke` is a categorical variable

smoke = if\_else(smoke == 9, NA,

if\_else(smoke == 1, "Active", "Inactive")))

**Part (c)**

library(ggplot2)

# Plotting observations whose smoking status was not missing

birth.complete.df <- birth.tidied.df %>% drop\_na()

# Attempt 1

ggplot(birth.complete.df, aes(x = gestation, y = bwt, colour = smoke)) +

geom\_point(alpha = 0.25) +

geom\_smooth(method = "lm", se = FALSE) + # Include naive best fit lines

theme\_bw() +

labs(x = "Gestation (days)", y = "Birth weight (ounces)", colour = "Smoking status",

title = "Baby birth weights versus gestation by mother's smoking status")

# Attempt 2

# Let's wrangle the data using approximate WHO classifications

birth.WHO.df <- birth.complete.df %>%

mutate(gestation\_WHO = if\_else(gestation < 37 \* 7, "Preterm",

if\_else(gestation < 36 \* 7, "Late preterm",

if\_else(gestation < 38 \* 7, "Early term",

if\_else(gestation < 40 \* 7, "Full term",

if\_else(gestation < 41 \* 7, "Late term",

"Postterm"))))),

gestation\_WHO = factor(gestation\_WHO, levels = c("Preterm", "Late preterm", "Early term",

"Full term", "Late term", "Postterm")))

ggplot(birth.WHO.df, aes(x = gestation\_WHO, y = bwt, fill = smoke)) +

geom\_boxplot() +

theme\_bw() +

labs(x = "Gestation (term)", y = "Birth weight (ounces)", fill = "Smoking status",

title = "Baby birth weights versus gestation by mother's smoking status")

It does seem that newborns of (active) smokers are smaller at every gestational age.

**NB:** No marks are deducted if the missing values of smoke were plotted.

**Part (d)**

# Attempt 1A

ggplot(birth.complete.df, aes(x = gestation, y = bwt, colour = smoke)) +

geom\_point(alpha = 0.5) +

geom\_smooth(method = "lm", se = FALSE) + # Include naive best fit lines

facet\_wrap( ~ age) +

theme\_bw() +

labs(x = "Gestation (days)", y = "Birth weight (ounces)", colour = "Smoking status",

title = "Baby birth weights versus gestation by mother's smoking status (facetted by mother's age)")

If we look within age, it does seem that newborns of (active) smokers are smaller at every gestational age. That said, it is difficult to discern if it is consistent with this visualisation.

**NB:** This plot is hard to read… Another attempt at this with material outside of what we have covered is below.

# Let's wrangle the data using groups of code defined age groups...

birth.bands <- birth.complete.df %>%

mutate(age\_group = cut(age, breaks = c(14, 20, 25, 30, 35, 40, 45),

labels = c("15–19 years", "20–24 years", "25–29 years",

"30–34 years", "35–39 years", "40–45 years")))

# Attempt 1B

ggplot(birth.bands, aes(x = gestation, y = bwt, colour = smoke)) +

geom\_point(alpha = 0.25) +

geom\_smooth(method = "lm", se = FALSE) + # Include naive best fit lines

facet\_wrap( ~ age\_group) +

theme\_bw() +

labs(x = "Gestation (term)", y = "Birth weight (ounces)", colour = "Smoking status",

title = "Baby birth weights versus gestation by mother's smoking status (facetted by mother's age)")

If we look within age bands instead, it is not as clear that newborns of (active) smokers are smaller at every gestational age. We may need formal statistical analyses to determine if age influences this relationship between birth weight and gestational age.

# Attempt 2

ggplot(birth.WHO.df, aes(x = gestation\_WHO, y = bwt, fill = smoke)) +

geom\_boxplot() +

facet\_wrap( ~ parity) +

theme\_bw() +

labs(x = "Gestation (days)", y = "Birth weight (ounces)", fill = "Smoking status",

title = "Baby birth weights versus gestation by mother's smoking status")

Excluding posterm newborns, it seems that newborns of (active) smokers are smaller at every gestational age, regardless of whether they were firstborns. However, for posterm, it only holds for newborns who were not firstborns.

**Question 5**

**Part (a)**

ranking.df <- read.csv("ranking.csv")

library(ggplot2); library(tidyr); library(dplyr)

ranking.long <- ranking.df %>%

pivot\_longer(cols = c("Best", "Worst"), names\_to = "Type", values\_to = "Rank") %>%

mutate(Type = if\_else(Type == "Best", "Best Rank", "Worst Rank")) %>%

arrange(Type, Rank) %>%

mutate(Team = factor(Team, levels = unique(Team)))

ggplot(ranking.long, aes(x = Rank, y = Team, group = Team)) +

geom\_path() +

geom\_point(aes(colour = Type), size = 3) +

labs(x = "World Rugby Rank", y = NULL, colour = NULL,

title = "Nations that have appeared in the Rubgy World Cup") +

scale\_x\_reverse() +

scale\_y\_discrete(limits = rev) +

theme\_bw() +

theme(text = element\_text(size = 16))

**Part (b)** The data is already in a tidy data format. But, we had to pivot it longer to plot a line between a country’s best and worst rank. From there, we sorted the data by the country’s best rank and enforced (alphabetical) best rank order using the factor() command. Now the data is in the appropriate form for plotting purposes.

We used the group input (argument) to ensure a single line was plotted per country. Then the best and worst ranks were plotted as colour-coded points. The “addition” of scale\_x\_reverse() flipped the order of the *x*-axis such that the best rank appeared on the right-hand side instead. Lastly, the “addition” of scale\_y\_discrete(limits = rev) flipped the order of the *y*-axis so the best ranks can be read from top-to-bottom. Lastly, the absence of the *y*-axis labels and the legend title for the colour-coded points were achieved by setting their labels to NULL.