Data Analysis Training Workshop

Grand Palm Hotel, Gaborone

16-20 May 2022

Presentation by

Dr John Grewar

jdgrewar@gmail.com

This Programme is funded by the European Unions (EU) & implemented by GOPA AFC JVL Consulting

Introduction

Workshop overview

- Installing R and RStudio
- GitHub and R in the cloud
- Using R for
  - Calculator
  - Statistics
  - Data depiction with graphs and maps
  - Simulate data (Risk analysis)
  - Reporting using Rmarkdown
- Other software – qGIS for clean maps

Workshop structure

1. Installing R and RStudio
2. GitHUB and working in the cloud
3. Introduction to R
    1. General
        1. Using functions
    2. Apply functions
4. Basic statistics
5. Frequency measures
6. Measures of association
7. Spatial data analysis
8. RMarkdown and reporting from R
9. Diagnostic test analysis

Available files

1. Full folder of all code and data for analysis – but we will be using GitHub to access this but in case there are issues you can use this directly
2. Tutorial videos where applicable
3. Software
    1. R
    2. RStudio
    3. Git for Windows
    4. qGIS

1. Installing R and RStudio

- **R and RStudio can be installed from online** at https://cloud.r-project.org/ and https://www.rstudio.com/products/rstudio/download/preview/ respectively but the software downloads are available in the software folder for the workshop - R-4.2.0-win.exe and RStudio-2022.02.1-461.exe respectively

2. GitHub and working in the cloud

Install GIT on your computer

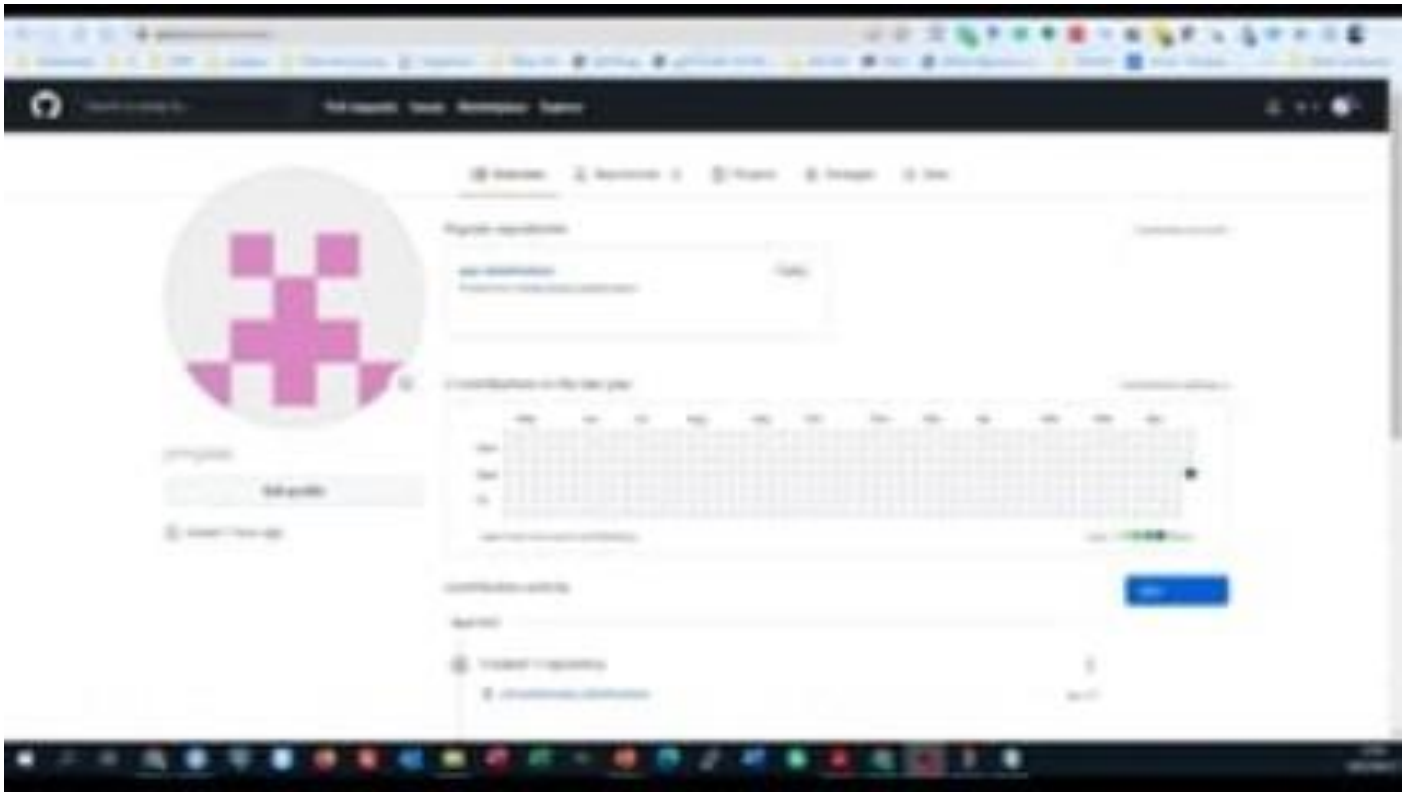
software files: Git-2.36.0-64-bit

or https://git-for-windows.github.io/

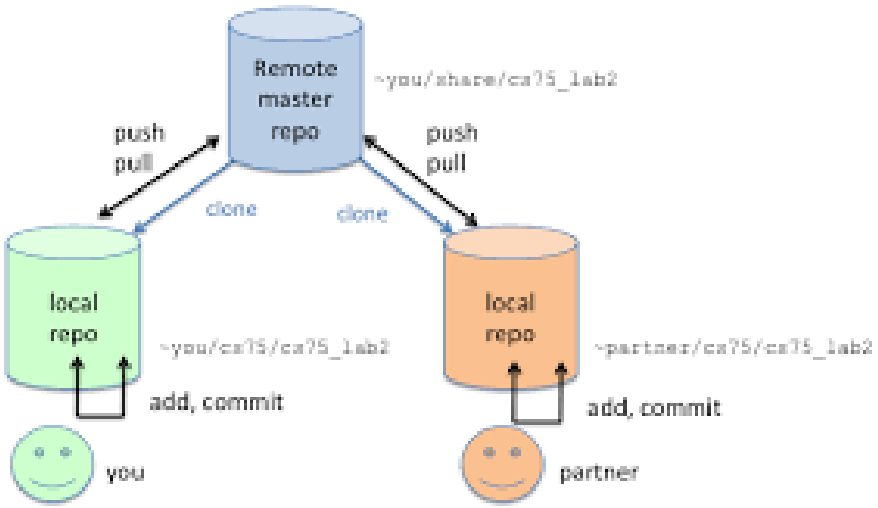
- Adjusting your PATH environment
  - SELECT *Git from the command line and also from 3rd-party software*
- RStudio for Windows prefers for Git to be installed below C:/Program Files and this appears to be the default

Create a GitHub account and fork the workshop repository – www.github.com

- **Create an authentication token**



DEMO – add a file and push to GitHub



Theme 1: Introduction - R; Vectors, Matrixes, Factors, Lists , Dataframes, Functions

1.1 RStudio and making a project file

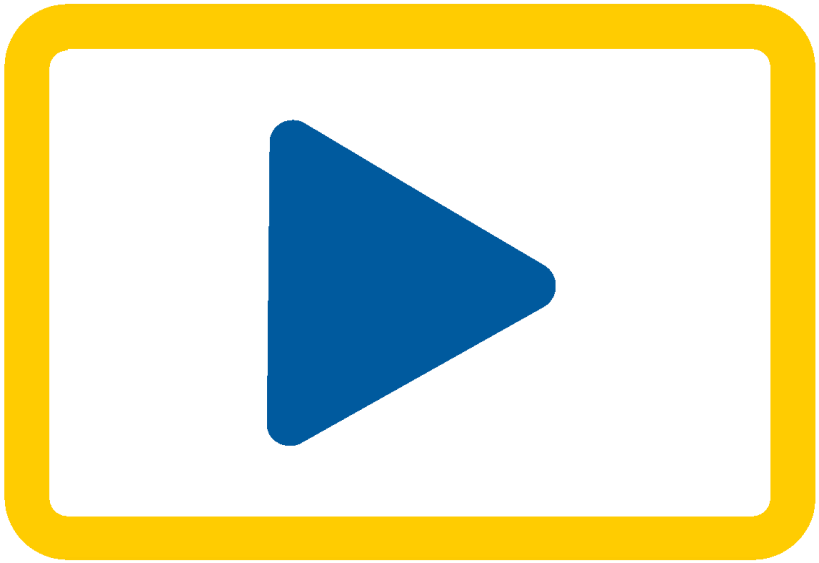
- Open and Navigate RStudio
- Open Project made in GitHub workflow
- Open 01_Introduction.R
- Check working directory

1.2 Introduction to R

- Run through the 01_Introduction.R file

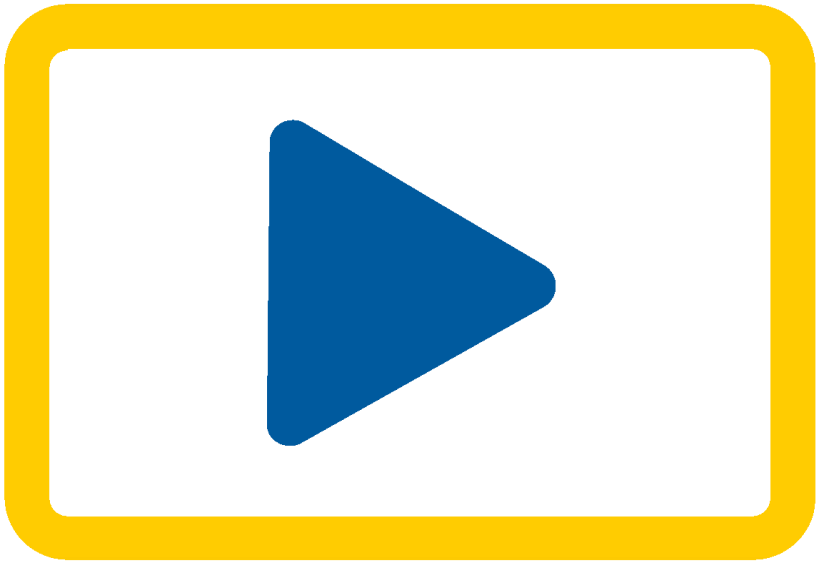
1.2 Introduction to R

- **Vector**
  - **Make vector of ages of attendees**
    - Name them
    - Summary stats
- **Dataframe (l 188)**
  - Create a dataframe based on the subset of wahid_species where the total number of cases are more than 10 AND the species is 'Cattle'

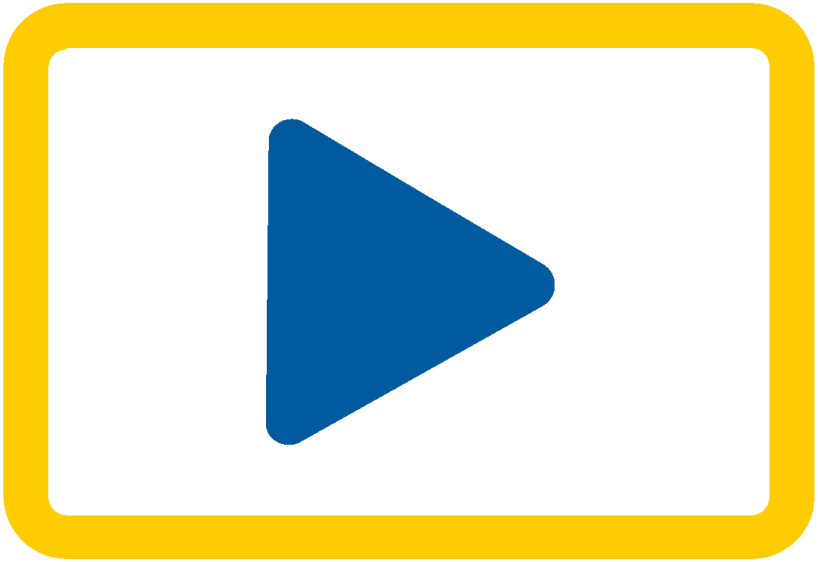


1.2 Introduction to R

- **Basic**
  - 1. create a vector of numbers of every even number between 2 and 20
  - 2. establish the square root of the sum of your number vector
  - 3. Create a vector of numbers between 1 and 50 (hint: use the : operator c(1:50))
  - 4. Create a matrix
- **Intermediate**
  - Use the sample function inbuilt in R to sample 10 numbers from the numbered vector in 3 above. hint: use the `?sample` command to take a look at the sample function



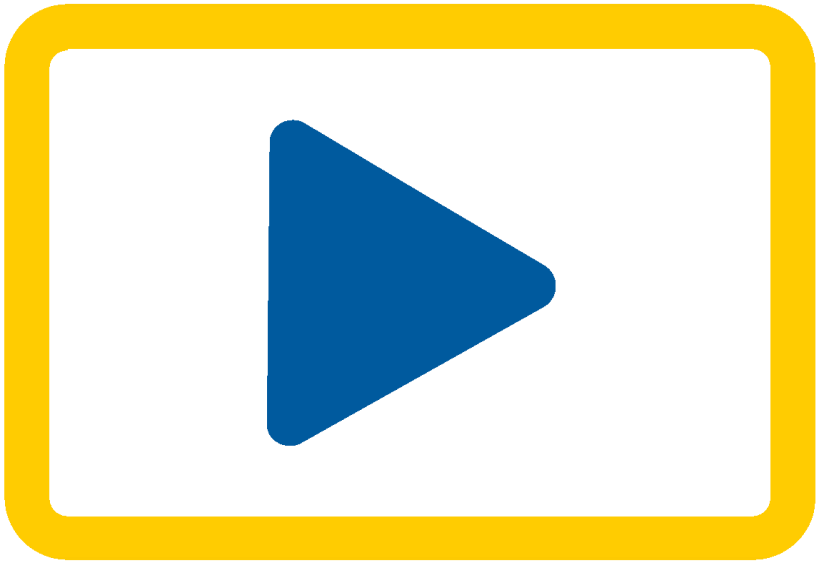
Commit and Push any changes or comments you have to your GitHub repository



- **Vectors (one dimensional array): can hold numeric, character or logical values.**
  - **All elements same data type**
- **Matrices (two dimensional array): can hold numeric, character or logical values.**
  - **All elements same data type**
- **Data frames (two-dimensional objects): can hold numeric, character or logical values.**
  - **Within each column the data-type is the same**

1.2 Introduction to R

- **Basic**
  - 1. create a vector of numbers of every even number between 1 and 20
  - 2. establish the square root of the sum of your number vector
  - 3. Create a vector of numbers between 1 and 50 (hint: use the : operator c(1:50))
  - 4. Create a matrix
- **Intermediate**
  - Use the sample function inbuilt in R to sample 10 numbers from the numbered vector in 3 above. hint: use the `?sample` command to take a look at the sample function



1.4 Introduction to R

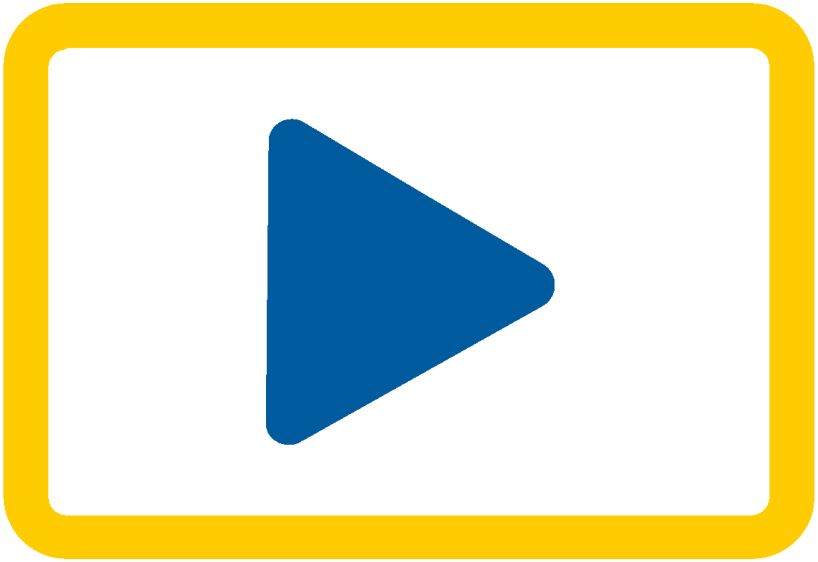
- Apply functions
- Open 01b_Apply.R

Theme 1: END

Theme 2: Basic statistics in R

- **Open 01_BasicStats.R**

Commit and Push any changes or comments you have to your GitHub repository



Theme 2: END

Theme 3: Measures of Disease Frequency

- **FMD sero-surveillance project in Masai Mara reserve in Kenya**
- **Philosophy of data and code sharing**

Preventive Veterinary Medicine 176 (2020) 104929

Contents lists available at ScienceDirect

Preventive Veterinary Medicine

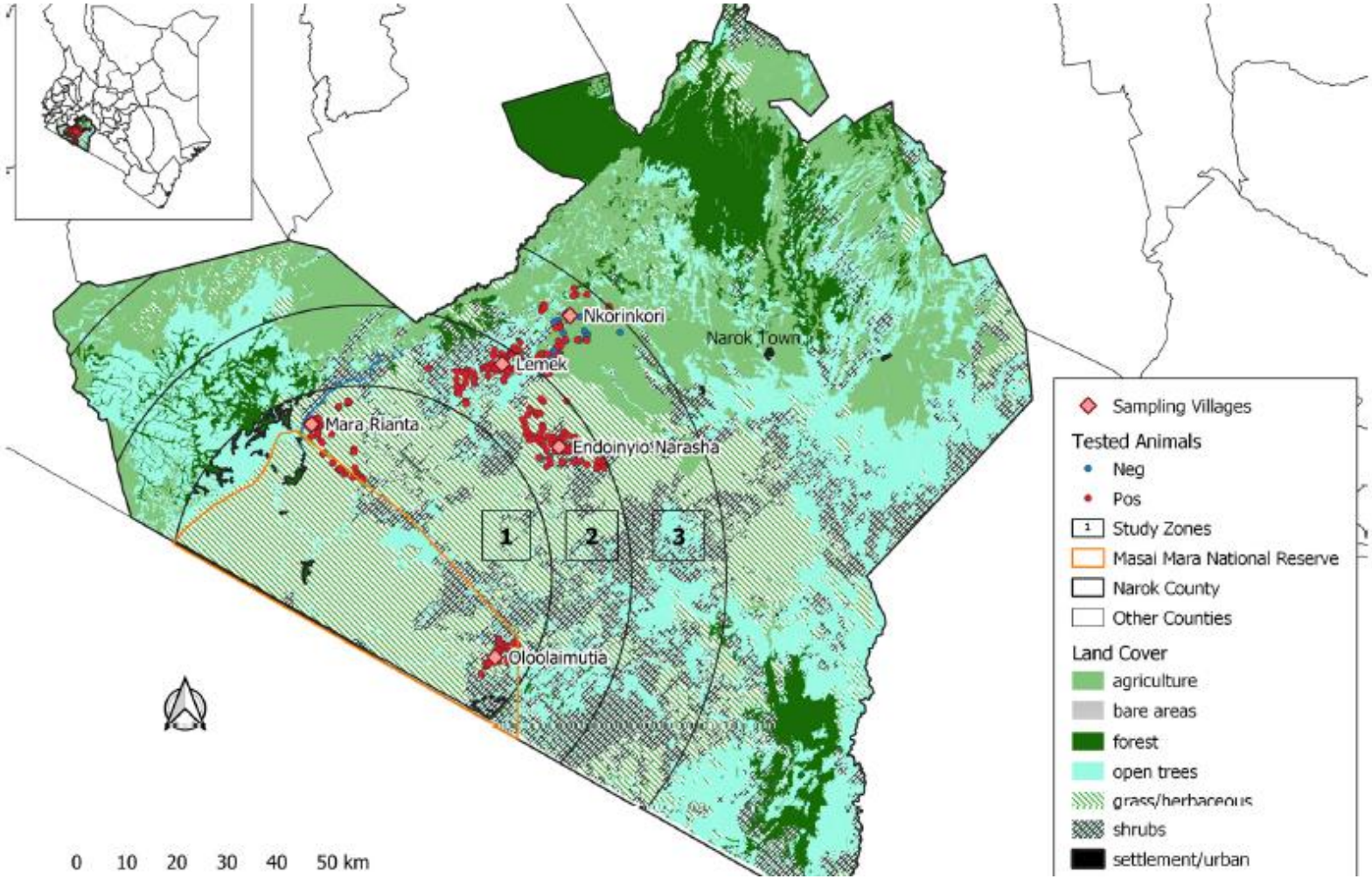
journal homepage: www.elsevier.com/locate/prevetmed

Seroprevalence of foot-and-mouth disease virus in cattle herds raised in Maasai Mara ecosystem in Kenya

Daniel Nthiwa[a,b,*], Bernard Bett[b], David Odongo[c], Eucharia Kenya[a], Martin Wainaina[b], Santina Grazioli[d], Efrem Foglia[d], Emiliana Brocchi[d], Silvia Alonso[e]

ABSTRACT

A cross-sectional study was carried out to determine foot-and-mouth disease (FMD) seroprevalence and identify risk factors of exposure among cattle herds raised in three zones with different types of land use and progressively distant from the Maasai Mara National Reserve (MMNR) boundary. We selected five villages purposively; two in zone 1 (area < 20 km from the MMNR), another two in zone 2 (area between 20 − 40 km away from the MMNR) and one in zone 3 (area > 40 km away from the MMNR). A total of 1170 cattle sera were collected from 390 herds in all the zones and tested for antibodies against the non-structural proteins (NSPs) of FMD virus (FMDV) using two 3ABC-based Enzyme-Linked Immunosorbent Assay ELISA kits. All sera samples were also screened for serotype-specific antibodies using Solid Phase Competitive ELISA (SPCE) kits (IZSLER, Italy). We targeted FMDV serotypes A, O, South African Territory [SAT] 1 and SAT 2, known to be endemic in East Africa including Kenya. Data on putative risk factors for FMD seropositivity in cattle were collected using a questionnaire. The overall apparent animal-level FMD seroprevalence based on the parallel comparison of the two anti-NSPs ELISA kits was 83.8 % (95 % CI; 81.8–85.9), and differed significantly across zones. Zone 1 had a higher seroprevalence than zones 2 and 3 ($\chi^2 = 116.1$, df = 2, $p < 0.001$). In decreasing order, the overall seroprevalences of FMDV serotypes A, SAT 2, O and SAT 1 were 26.3 % (95 % CI; 23.5–29.2), 21.4 % (95 % CI; 18.8–24.0), 21.2 % (95 % CI; 18.7–23.9) and 13.1 % (95 % CI; 11.1–15.3), respectively. The distribution of these serotypes differed significantly between zones ($p < 0.05$) except for SAT 2 serotype ($\chi^2 = 0.90$, df = 2, $p = 0.639$). Both serotypes A and O were more prevalent in zones 1 and 2 than zone 3 while serotype SAT 1, was higher in zone 3 compared to other zones. The results of multivariable analyses identified animal sex (i.e., female), raising of cattle in zones 1 and 2 (areas < 40 km away from the MMNR); mixing of cattle from multiple herds at watering points, and pastoral husbandry practices, as significant predictors of animal-level FMD ser-


Nkorinkori
Narok Town
Lemek
Mara Rianta
Endoinyio Narasha
1
2
3
Oloolaimutia
0 10 20 30 40 50 km
Sampling Villages
Tested Animals
Neg
Pos
Study Zones
Masai Mara National Reserve
Narok County
Other Counties
Land Cover
agriculture
bare areas
forest
open trees
grass/herbaceous
shrubs
settlement/urban

- **Open 03_Frequency.R**

3.1 Results

- **Total tested**
- **Sex proportions**
- **Total herds tested**
- **Apparent prevalence with confidence intervals**

3. Results

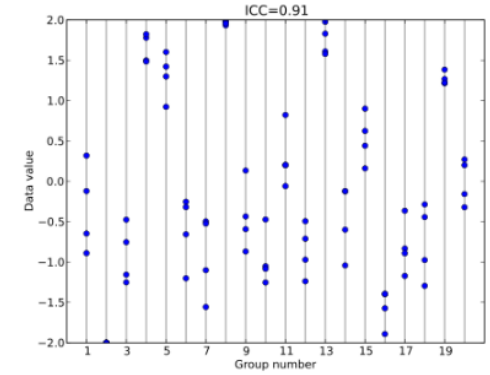
3.1. Anti-NSP antibodies prevalence and distribution

In total 1,170 cattle sera (78.6% female and 21.4% male) from 390 herds were tested for antibodies against NSPs using two ELISA assays. The proportion of sampled herds described as vaccinated at the time of sampling was 44.9 %. The overall apparent animal-level and true seroprevalences of FMD were 83.8 % (95 % CI; 81.5–86.2) and 84.7 % (95 % CI; 82.4–86.9), respectively. The apparent animal-level seroprevalence of FMD differed significantly between locations where animals were kept. Zone 1 had a higher seroprevalence compared to zones 2 and 3 ($\chi^2 = 116.1$, df = 2, $p < 0.001$) (Table 1). The spatial distribution of NSP-positive animals in the surveyed zones is shown in Fig. 1. FMD animal-level seroprevalence also differed significantly by sex ($\chi^2 = 14.5$, df = 1, $p < 0.001$), with more female animals (86.0 %; 95 % CI; 83.8–88.2) being seropositive than males (76.0 %; 95 % CI; 70.8–81.2) (Table 1). The level of agreement between both ELISA tests was moderate (Cohen's Kappa statistic k = 0.6). The diagnostic sensitivity of both ELISA tests differed significantly (McNemar's $\chi^2 = 60.9$, df = 1, $p < 0.001$); the PrioCHECK® FMDV NS ELISA test detected more NSP positives, 81.2 % (95 % CI; 78.7–83.7) than the IZSLER in-house 3ABC trapping ELISA, 72.3 % (69.5–75.1).

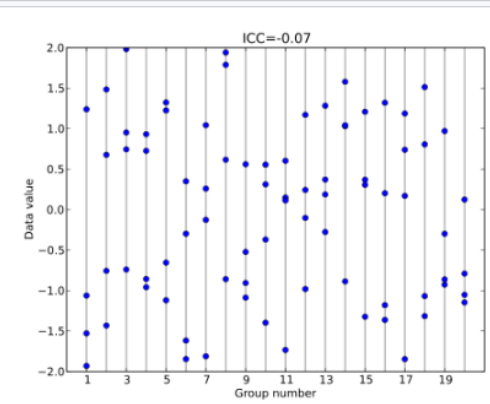
3.1.4 – Intraclass Correlation

- How strongly do units/animals in the same groups resemble each other
- [0,1]
- 0 = perfectly independent
- 1 = perfectly dependent
- Consider diseases like FMD
- Consider diseases like BSE

https://en.wikipedia.org/wiki/Intraclass_correlation



A dot plot showing a dataset with high intraclass correlation. Values from the same group tend to be similar.



A dot plot showing a dataset with low intraclass correlation. There is no tendency for values from the same group to be similar.

3.1.4 – Intraclass Correlation

- Impacts sample size BECAUSE
  - Impacts outcome of prevalence survey
- 0.1 is pretty low – the outcomes of the study showed an ICC of 0.24
- An increasing ICC (ρ) Rho will increase sample size
- An increase sample size per herd will increase sample size
- In this study the ability to sample so few animals per herd made the ICC impact small

limited in the area. We accounted for design effect due to herd-level clustering of cattle by adjusting the initial sample size using the formula; $n^1 = n(1 + \rho(m-1))$, where n^1 is the adjusted sample size, ρ the intra-cluster (within-herd) correlation coefficient (ICC), and m the number of animals to be sampled in each herd (Dohoo et al., 2012). The study assumed an ICC of 0.1 for FMD and sampled 3 animals (randomly-selected) in each herd. A total of 465 animals (from 155 herds)

3.1.4 Task

- Create a function in R to work out the design effect given a value of ICC (ρ) and the number of animals sampled per herd (m)
- Call the function *deff*

Sample size

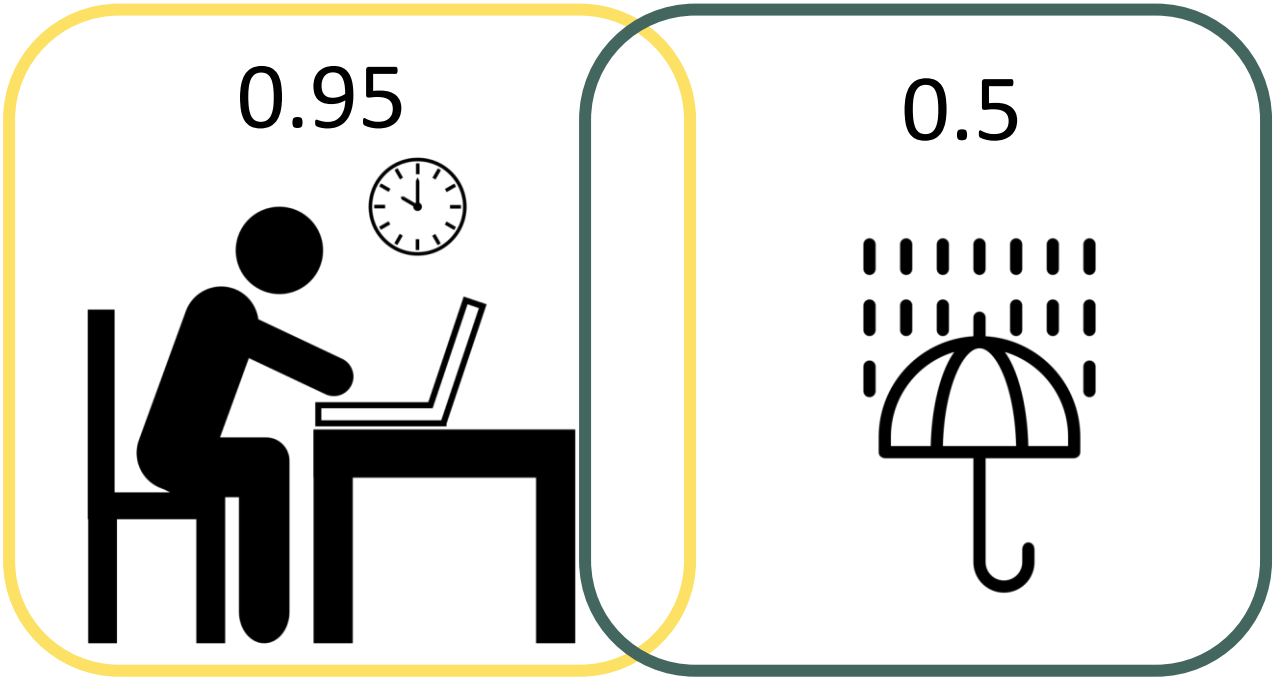
$$n' = n\left(1 + \rho(m-1)\right)$$

Design effect (DEFF)

$$\frac{n'}{n}$$

DEFF = $1 + \rho(m-1)$

Probability



On any given day what's the probability of either happening? (i.e. Parallel)

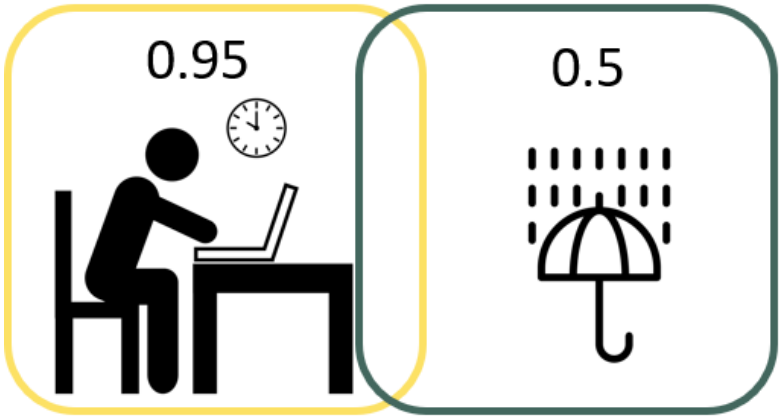
$$Pwork + Prain$$

0.95 + 0.5 = 1.45 …

$$Pwork + Prain - (Pwork * Prain)$$

0.95 + 0.5 – (0.95x0.5)

= 0.975 = 97.5% chance that you will work or it will rain



$$Se_{||} = Se_1 + Se_2 - (Se_1 * Se_2)$$

Bonus: What it's the probability that it will both rain and you will work on any given day?

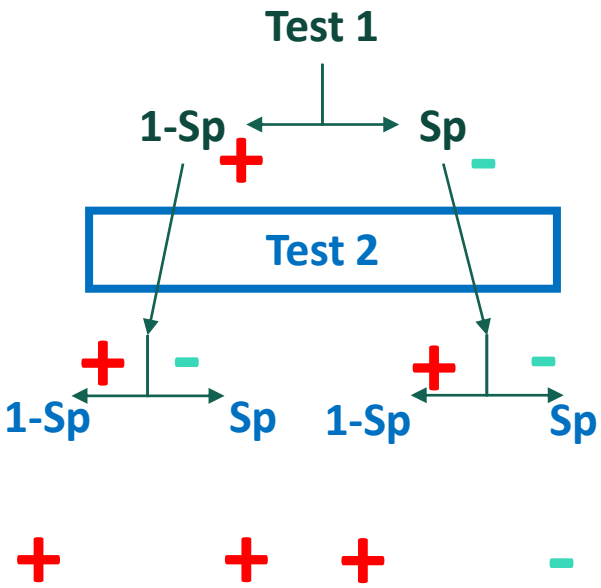
3.1.4b: Interpreting diagnostic tests in Parallel

1. Sensitivity

$$Se_{||} = Se_1 + Se_2 - (Se_1 * Se_2)$$

2. Specificity

NEGATIVE SAMPLE



3.1.4b: Interpreting diagnostic tests in Parallel

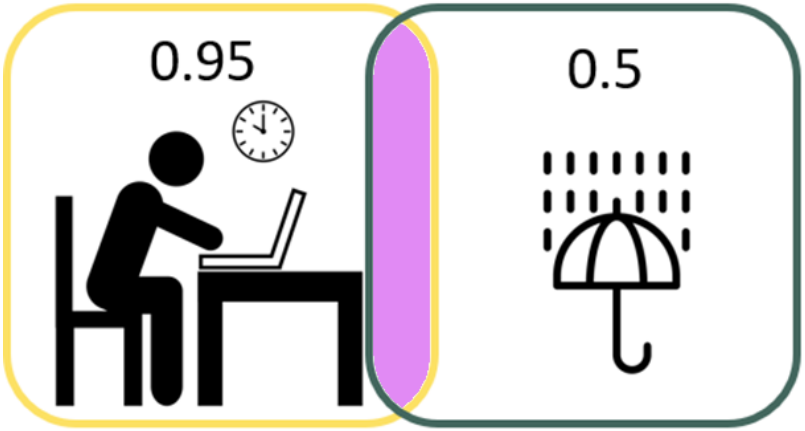
1. **Sensitivity**

$$Se_{||} = Se_1 + Se_2 - (Se_1 * Se_2)$$

2. **Specificity**

$$Sp_{||} = Sp_1 * Sp_2$$

testing in parallel increases sensitivity but decreases specificity



3.1.6 Task establish the sex proportions with their 95% confidence intervals

Variable and category		FMD
	No. tested (n)	% NSP prevalence (95 % CI)
Sex		
Male	250	76.0 (70.1–81.8)
Female	920	86.0 (83.5–88.4)

3.1.7 - Prevalence estimates - Botswana context

- Goal - extract all foot-and-mouth disease data
- Assume design effect of 1 (no clustering between sampled herds)
- Plot these prevalence's using `ggplot`
- Facet these graphs by species

3.1.7.2

Data visualization with ggplot2 : : CHEAT SHEET

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data** set, a **coordinate system**, and **geoms**—visual marks that represent data points.

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

Complete the template below to build a graph.

```
ggplot (data = <DATA>) +                                   required
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
 stat = <STAT>, position = <POSITION>) +                   Not required, sensible defaults supplied
<COORDINATE_FUNCTION> +
<FACET_FUNCTION> +
<SCALE_FUNCTION> +
<THEME_FUNCTION>
```

ggplot(data = mpg, aes(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

last_plot() Returns the last plot.

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Aes

Common aesthetic values.

color and **fill** - string ("red", "#RRGGBB")

linetype - integer or string (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "dotdash", 5 = "longdash", 6 = "twodash")

lineend - string ("round", "butt", or "square")

linejoin - string ("round", "mitre", or "bevel")

size - integer (line width in mm)

shape - integer/shape name or a single character ("a")

RStudio

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank() and **a + expand_limits()** Ensure limits include values across all plots.

b + geom_curve(aes(yend = lat + 1, xend = long + 1), curvature = 1) - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path(lineend = "butt", linejoin = "round", linemitre = 1) x, y, alpha, color, group, linetype, size

a + geom_polygon(aes(alpha = 50)) - x, y, alpha, color, fill, group, subgroup, linetype, size

b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900)) - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + geom_abline(aes(intercept = 0, slope = 1))
b + geom_hline(aes(yintercept = lat))
b + geom_vline(aes(xintercept = long))

b + geom_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:1155, radius = 1))

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin") x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian") x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot() x, y, alpha, color, fill

c + geom_freqpoly() x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)) x, y, alpha, color, fill, linetype, size, weight

discrete

d <- ggplot(mpg, aes(fl))

d + geom_bar() x, alpha, color, fill, linetype, size, weight

TWO VARIABLES both continuous

e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_point() x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile() x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl") x, y, alpha, color, linetype, size

e + geom_smooth(method = lm) x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

one discrete, one continuous

f <- ggplot(mpg, aes(class, hwy))

f + geom_col() x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot() x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center") x, y, alpha, color, fill, group

f + geom_violin(scale = "area") x, y, alpha, color, fill, group, linetype, size, weight

both discrete

g <- ggplot(diamonds, aes(cut, color))

g + geom_count() x, y, alpha, color, fill, shape, size, stroke

e + geom_jitter(height = 2, width = 2) x, y, alpha, color, fill, shape, size

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500)) x, y, alpha, color, fill, linetype, size, weight

h + geom_density_2d() x, y, alpha, color, group, linetype, size

h + geom_hex() x, y, alpha, color, fill, size

continuous function

i <- ggplot(economics, aes(date, unemploy))

i + geom_area() x, y, alpha, color, fill, linetype, size

i + geom_line() x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv") x, y, alpha, color, group, linetype, size

visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))

j + geom_crossbar(fatten = 2) - x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar() - x, ymax, ymin, alpha, color, group, linetype, size, width Also **geom_errorbarh**().

j + geom_linerange() x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange() - x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))

k + geom_map(aes(map_id = state), map = map) + **expand_limits**(x = map$long, y = map$lat) map_id, alpha, color, fill, linetype, size

THREE VARIABLES

seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); l <- ggplot(seals, aes(long, lat))

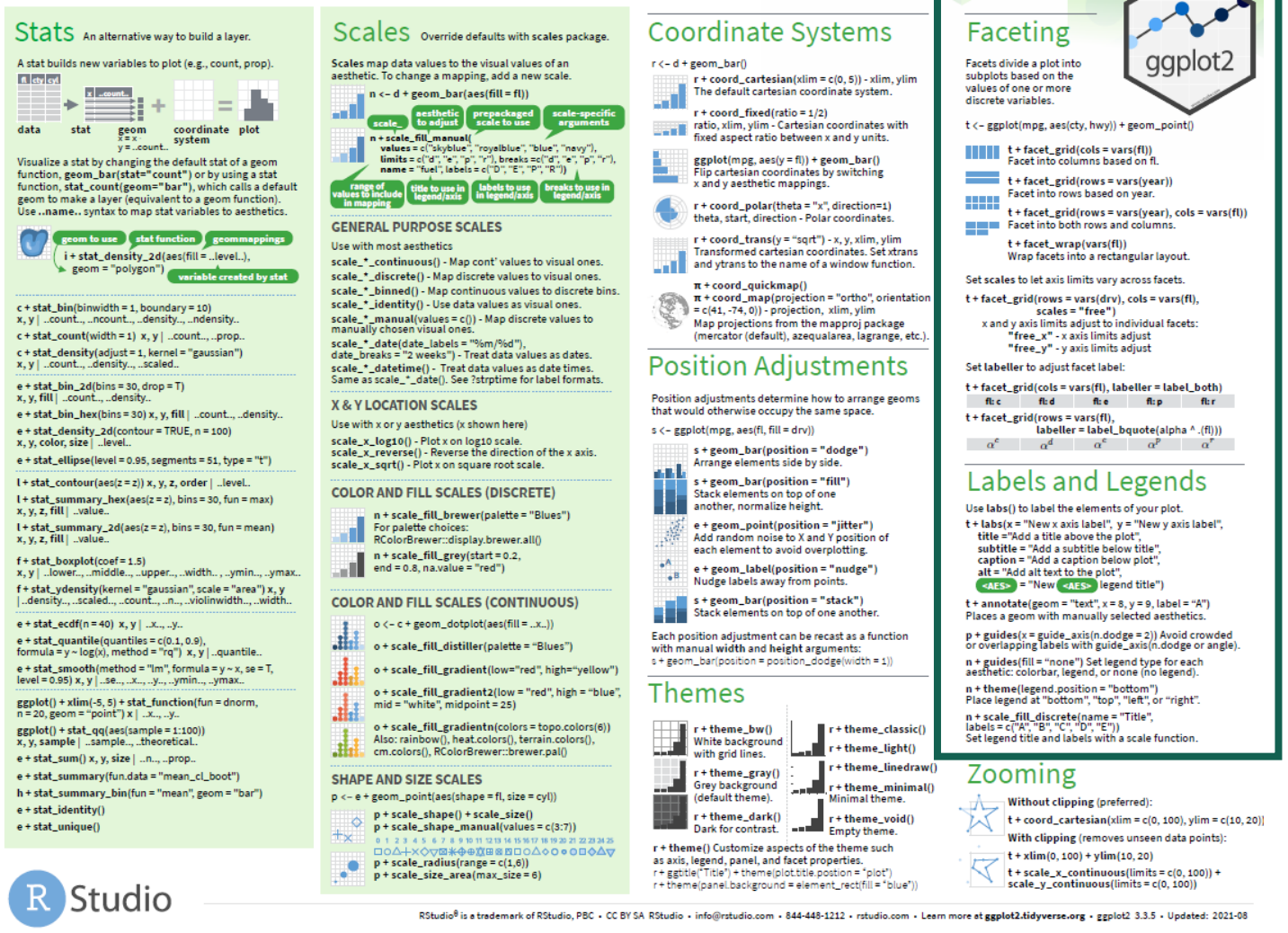
l + geom_contour(aes(z = z)) x, y, z, alpha, color, group, linetype, size, weight

l + geom_contour_filled(aes(fill = z)) x, y, alpha, color, fill, group, linetype, size, subgroup

l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE) x, y, alpha, fill

l + geom_tile(aes(fill = z)) x, y, alpha, color, fill, linetype, size, width

RStudio® is a trademark of RStudio, PBC • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at **ggplot2.tidyverse.org** • ggplot2 3.3.5 • Updated: 2021-08



Stats An alternative way to build a layer.

A stat builds new variables to plot (e.g., count, prop).

data → stat → geom → coordinate system → plot

Visualize a stat by changing the default stat of a geom function, **geom_bar(stat="count")** or by using a stat function, **stat_count(geom="bar")**, which calls a default geom to make a layer (equivalent to a geom function). Use **..name..** syntax to map stat variables to aesthetics.

geom to use | stat function | geommappings

i + stat_density_2d(aes(fill = ..level..), geom = "polygon")

variable created by stat

c + stat_bin(binwidth = 1, boundary = 10)
x, y | ..count.., ..ncount.., ..density.., ..ndensity..

c + stat_count(width = 1) x, y | ..count.., ..prop..

c + stat_density(adjust = 1, kernel = "gaussian")
x, y | ..count.., ..density.., ..scaled..

e + stat_bin_2d(bins = 30, drop = T)
x, y, fill | ..count.., ..density..

e + stat_bin_hex(bins = 30) x, y, fill | ..count.., ..density..

e + stat_density_2d(contour = TRUE, n = 100)
x, y, color, size | ..level..

e + stat_ellipse(level = 0.95, segments = 51, type = "t")

l + stat_contour(aes(z = z)) x, y, z, order | ..level..

l + stat_summary_hex(aes(z = z), bins = 30, fun = max)
x, y, z, fill | ..value..

l + stat_summary_2d(aes(z = z), bins = 30, fun = mean)
x, y, z, fill | ..value..

f + stat_boxplot(coef = 1.5)
x, y | ..lower.., ..middle.., ..upper.., ..width.. , ..ymin.., ..ymax..

f + stat_ydensity(kernel = "gaussian", scale = "area") x, y
| ..density.., ..scaled.., ..count.., ..n.., ..violinwidth.., ..width..

e + stat_ecdf(n = 40) x, y | ..x.., ..y..

e + stat_quantile(quantiles = c(0.1, 0.9),
formula = y ~ log(x), method = "rq") x, y | ..quantile..

e + stat_smooth(method = "lm", formula = y ~ x, se = T,
level = 0.95) x, y | ..se.., ..x.., ..y.., ..ymin.., ..ymax..

ggplot() + xlim(-5, 5) + **stat_function**(fun = dnorm,
n = 20, geom = "point") x | ..x.., ..y..

ggplot() + stat_qq(aes(sample = 1:100))
x, y, sample | ..sample.., ..theoretical..

e + stat_sum() x, y, size | ..n.., ..prop..

e + stat_summary(fun.data = "mean_cl_boot")

h + stat_summary_bin(fun = "mean", geom = "bar")

e + stat_identity()

e + stat_unique()

Scales Override defaults with scales package.

Scales map data values to the visual values of an aesthetic. To change a mapping, add a new scale.

n <- d + geom_bar(aes(fill = fl))

scale_ | aesthetic to adjust | prepackaged scale to use | scale-specific arguments

n + scale_fill_manual(
values = c("skyblue", "royalblue", "blue", "navy"),
limits = c("d", "e", "p", "r"), breaks =c("d", "e", "p", "r"),
name = "fuel", labels = c("D", "E", "P", "R"))

range of values to include in mapping | title to use in legend/axis | labels to use in legend/axis | breaks to use in legend/axis

GENERAL PURPOSE SCALES

Use with most aesthetics

scale_*_continuous() - Map cont' values to visual ones.
scale_*_discrete() - Map discrete values to visual ones.
scale_*_binned() - Map continuous values to discrete bins.
scale_*_identity() - Use data values as visual ones.
scale_*_manual(values = c()) - Map discrete values to manually chosen visual ones.
scale_*_date(date_labels = "%m/%d"),
date_breaks = "2 weeks") - Treat data values as dates.
scale_*_datetime() - Treat data values as date times.
Same as scale_*_date(). See ?strptime for label formats.

X & Y LOCATION SCALES

Use with x or y aesthetics (x shown here)

scale_x_log10() - Plot x on log10 scale.
scale_x_reverse() - Reverse the direction of the x axis.
scale_x_sqrt() - Plot x on square root scale.

COLOR AND FILL SCALES (DISCRETE)

n + scale_fill_brewer(palette = "Blues")
For palette choices:
RColorBrewer::display.brewer.all()

n + scale_fill_grey(start = 0.2,
end = 0.8, na.value = "red")

COLOR AND FILL SCALES (CONTINUOUS)

o <- c + geom_dotplot(aes(fill = ..x..))

o + scale_fill_distiller(palette = "Blues")

o + scale_fill_gradient(low="red", high="yellow")

o + scale_fill_gradient2(low = "red", high = "blue",
mid = "white", midpoint = 25)

o + scale_fill_gradientn(colors = topo.colors(6))
Also: rainbow(), heat.colors(), terrain.colors(),
cm.colors(), RColorBrewer::brewer.pal()

SHAPE AND SIZE SCALES

p <- e + geom_point(aes(shape = fl, size = cyl))

p + scale_shape() + **scale_size**()
p + scale_shape_manual(values = c(3:7))

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

p + scale_radius(range = c(1,6))
p + scale_size_area(max_size = 6)

Coordinate Systems

r <- d + geom_bar()

r + coord_cartesian(xlim = c(0, 5)) - xlim, ylim
The default cartesian coordinate system.

r + coord_fixed(ratio = 1/2)
ratio, xlim, ylim - Cartesian coordinates with fixed aspect ratio between x and y units.

ggplot(mpg, aes(y = fl)) + **geom_bar**()
Flip cartesian coordinates by switching x and y aesthetic mappings.

r + coord_polar(theta = "x", direction=1)
theta, start, direction - Polar coordinates.

r + coord_trans(y = "sqrt") - x, y, xlim, ylim
Transformed cartesian coordinates. Set xtrans and ytrans to the name of a window function.

π + coord_quickmap()
π + coord_map(projection = "ortho", orientation = c(41, -74, 0)) - projection, xlim, ylim
Map projections from the mapproj package (mercator (default), azequalarea, lagrange, etc.).

Position Adjustments

Position adjustments determine how to arrange geoms that would otherwise occupy the same space.

s <- ggplot(mpg, aes(fl, fill = drv))

s + geom_bar(position = "dodge")
Arrange elements side by side.

s + geom_bar(position = "fill")
Stack elements on top of one another, normalize height.

e + geom_point(position = "jitter")
Add random noise to X and Y position of each element to avoid overplotting.

e + geom_label(position = "nudge")
Nudge labels away from points.

s + geom_bar(position = "stack")
Stack elements on top of one another.

Each position adjustment can be recast as a function with manual **width** and **height** arguments:
s + geom_bar(position = position_dodge(width = 1))

Themes

r + theme_bw()
White background with grid lines.

r + theme_gray()
Grey background (default theme).

r + theme_dark()
Dark for contrast.

r + theme_classic()

r + theme_light()

r + theme_linedraw()

r + theme_minimal()
Minimal theme.

r + theme_void()
Empty theme.

r + theme() Customize aspects of the theme such as axis, legend, panel, and facet properties.
r + ggtitle("Title") + theme(plot.title.position = "plot")
r + theme(panel.background = element_rect(fill = "blue"))

Faceting

Facets divide a plot into subplots based on the values of one or more discrete variables.

t <- ggplot(mpg, aes(cty, hwy)) + **geom_point**()

t + facet_grid(cols = vars(fl))
Facet into columns based on fl.

t + facet_grid(rows = vars(year))
Facet into rows based on year.

t + facet_grid(rows = vars(year), cols = vars(fl))
Facet into both rows and columns.

t + facet_wrap(vars(fl))
Wrap facets into a rectangular layout.

Set **scales** to let axis limits vary across facets.

t + facet_grid(rows = vars(drv), cols = vars(fl),
scales = "free")
x and y axis limits adjust to individual facets:
"free_x" - x axis limits adjust
"free_y" - y axis limits adjust

Set **labeller** to adjust facet label:

t + facet_grid(cols = vars(fl), **labeller = label_both**)

fl: c	fl: d	fl: e	fl: p	fl: r

t + facet_grid(rows = vars(fl),
labeller = label_bquote(alpha ^ .(fl)))

α^c	α^d	α^e	α^p	α^r

Labels and Legends

Use **labs()** to label the elements of your plot.

t + labs(x = "New x axis label", y = "New y axis label",
title ="Add a title above the plot",
subtitle = "Add a subtitle below title",
caption = "Add a caption below plot",
alt = "Add alt text to the plot",
<AES> = "New <AES> legend title")

t + annotate(geom = "text", x = 8, y = 9, label = "A")
Places a geom with manually selected aesthetics.

p + guides(x = guide_axis(n.dodge = 2)) Avoid crowded or overlapping labels with guide_axis(n.dodge or angle).

n + guides(fill = "none") Set legend type for each aesthetic: colorbar, legend, or none (no legend).

n + theme(legend.position = "bottom")
Place legend at "bottom", "top", "left", or "right".

n + scale_fill_discrete(name = "Title",
labels = c("A", "B", "C", "D", "E"))
Set legend title and labels with a scale function.

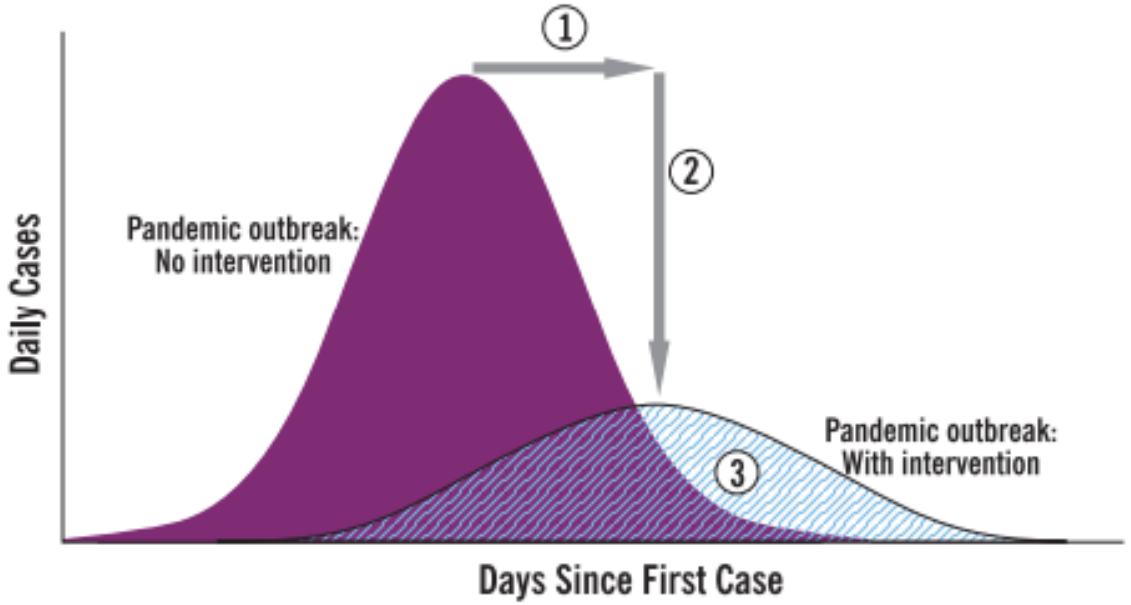
Zooming

Without clipping (preferred):
t + coord_cartesian(xlim = c(0, 100), ylim = c(10, 20))
With clipping (removes unseen data points):
t + xlim(0, 100) + **ylim**(10, 20)
t + scale_x_continuous(limits = c(0, 100)) +
scale_y_continuous(limits = c(0, 100))

RStudio® is a trademark of RStudio, PBC • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at **ggplot2.tidyverse.org** • ggplot2 3.3.5 • Updated: 2021-08

3.2 Frequency over time - epidemic curves

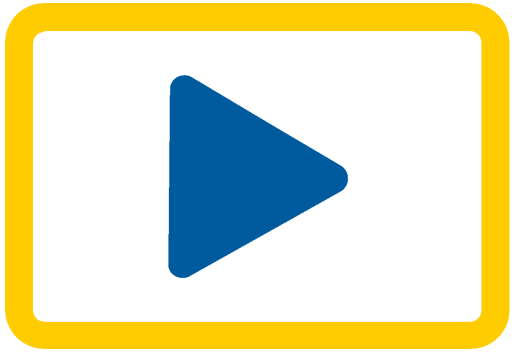
one of the most common, and important portrayals of animal (all) health data



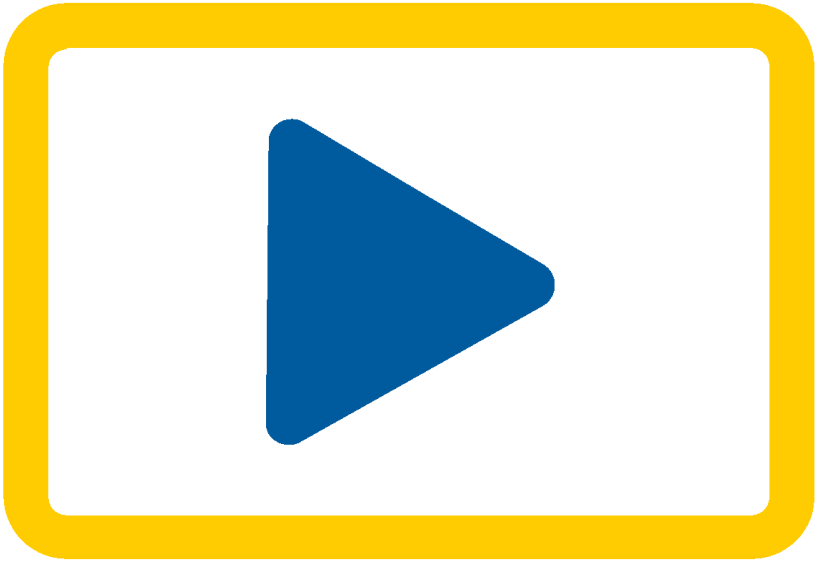
CDC: https://stacks.cdc.gov/view/cdc/11425

Task: lets redo those Epidemic curve graphs but focusing on FMD - but do this without changing the underlying data, just use the ggplot language

Can you go one step further and just look at 2014 and 2015? – for this you might want to change the underlying data



Commit and Push any changes or comments you have to your GitHub repository



Theme 3: END

Theme 4: Measures of Association

Table 1
Variables associated with animal-level seroprevalence of FMD based on univariable logistic regression with random effect for herd ID.

Variable and category		FMD		
	No. tested (n)	% NSP prevalence (95 % CI)	Odds ratio (95 % CI)	P-value
Sex				
Male	250	76.0 (70.1–81.8)	1 (Ref.)	
Female	920	86.0 (83.5–88.4)	2.9 (1.8–4.7)	< 0.001
Zones*				
Zone 3	240	63.3 (57.5–69.9)	1 (Ref.)	
Zone 2	465	83.4 (79.7–87.2)	3.6 (2.2–5.9)	< 0.001
Zone 1	465	94.8 (92.6–97.0)	14.7 (7.8–27.7)	< 0.001

Interpret regression coefficients

- Categorical (as in our example)
  - Accounting for other variables in the model (not what they have presented but relevant), the odds of female animals being sero-positive for FMD were 2.9 times higher when compared to male animals
  - Odds ratio without accounting for other variables in this case is 1.93 (we will perform this in R – in a 2X2 table the $OR = AD/BC$)

Sex				
Male	250	76.0 (70.1–81.8)	1 (Ref.)	
Female	920	86.0 (83.5–88.4)	2.9 (1.8–4.7)	< 0.001
Zones*				
Zone 3	240	63.3 (57.5–69.9)	1 (Ref.)	
Zone 2	465	83.4 (79.7–87.2)	3.6 (2.2–5.9)	< 0.001
Zone 1	465	94.8 (92.6–97.0)	14.7 (7.8–27.7)	< 0.001

Fixed and Random effects

Fixed effects are variables that are constant across individuals; these variables, like age, sex, or ethnicity, don't change or change at a constant rate over time

Random effects are those variables where there may be variance in outcomes based on groupings – commonly in Veterinary Epidemiology it may be as a result of herds.

- Like ICC – it is a way of accounting for uncertainty of within herd variance

- $glmer(fmd_\exp_st \sim anim_sex + (1|hse_id)$

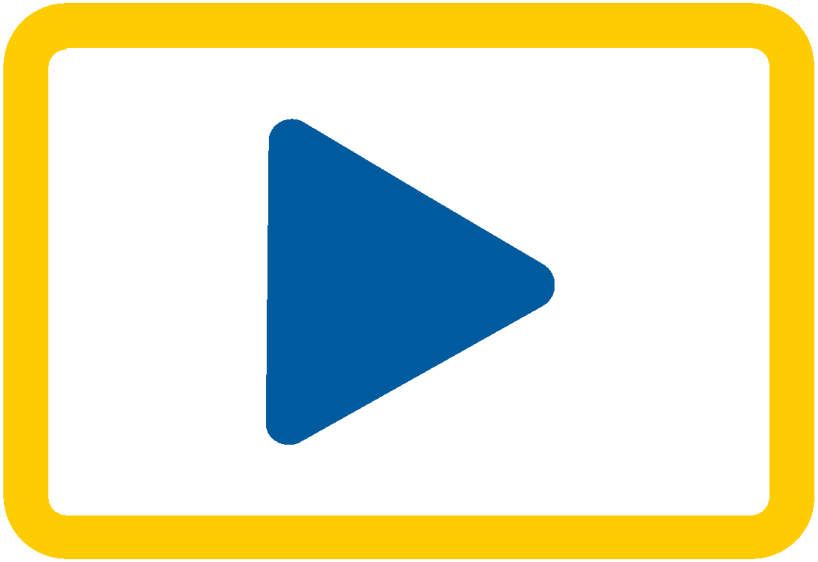
- **Open 04_MeasuresOfAssociation.R**

epiR::epi.2by2

The order for the functionality defines the output

	Disease +	Disease -	
Exposure + (test +)	**A**	**B**	TOTAL Exposed
Exposure – (test -)	**C**	**D**	TOTAL Unexposed
	TOTAL Diseased	TOTAL Free	TOTAL

4.2 Perform univariable logistic regression accounting for Herd as a Fixed effect for two of the other variables in the Nthiwa et al. dataset



4.3 A chi-square ($\chi 2$) statistic is a measure of the difference between the observed and expected frequencies of the outcomes of a set of events or variables

Typically used when you have a X by 2 setup

Use fishers exact test when any value in the table is 5 or below

4.4 Multivariate analysis

- Accounting for other variables in the data
- Notice the impact on the sex variable – shows the impact that considering other variables has

Table 3
Final multivariable model of animal-level risk factors for FMD in cattle based on GLMM analysis with a random effect for herd ID.

Variables	Category	Odds ratio (95 % CI)	P-*value*
Fixed effects			
Animal sex			
	Male	1 (Ref.)	
	Female	4.0 (2.5–6.4)	< 0.001
Study zones*			
	Zone 3	1 (Ref.)	
	Zone 2	3.2 (1.9–5.5)	< 0.001
	Zone 1	6.6 (2.2–19.6)	0.001
Contact with cattle (from a different herd) at watering points			
	No	1 (Ref.)	
	Yes	1.7 (1.0–2.8)	0.045
Herd management practice			
	Sedentary	1 (Ref.)	
	Pastoral	2.6 (1.0–6.8)	0.049

Theme 4: END

Theme 5: Spatial Analysis

- **Replicating the analysis and plot in Nthiwa et al.**
- **Difference will be the use of a Cattle density layer rather than land use**

5.A Aside

1. importing tabular excel files

```
importTabularOnlineExcel<- function(
  baseURL. = baseURL,
  folder = "tabular",
  filePlusExt = NULL
  ){

  if (missing(filePlusExt)) {
    stop("Please provide a vaild file to be sourced")
  }

temp = tempfile()
download.file(paste0(baseURL., "/",folder,"/",filePlusExt), destfile=temp, mode='wb')
return(data.frame(readxl::read_excel(temp)))
unlink(temp)
}
```

```
importShapefilesOnline<- function(
  baseURL. = baseURL,
  folder = "gis/shapefiles",
  filePlusExt = NULL
){

  if (missing(filePlusExt)) {
    stop("Please provide a vaild file to be sourced")
  }
  library(terra)
  temp <- tempfile()
  tempunzipped <- tempfile()
  URL = paste0(baseURL.,"/", folder,"/", filePlusExt)
  download.file(URL, temp)
  unzip(zipfile = temp, exdir = tempunzipped)
  return(vect(tempunzipped))
  unlink(temp)
  unlink(tempunzipped)
}
```

5.A Aside

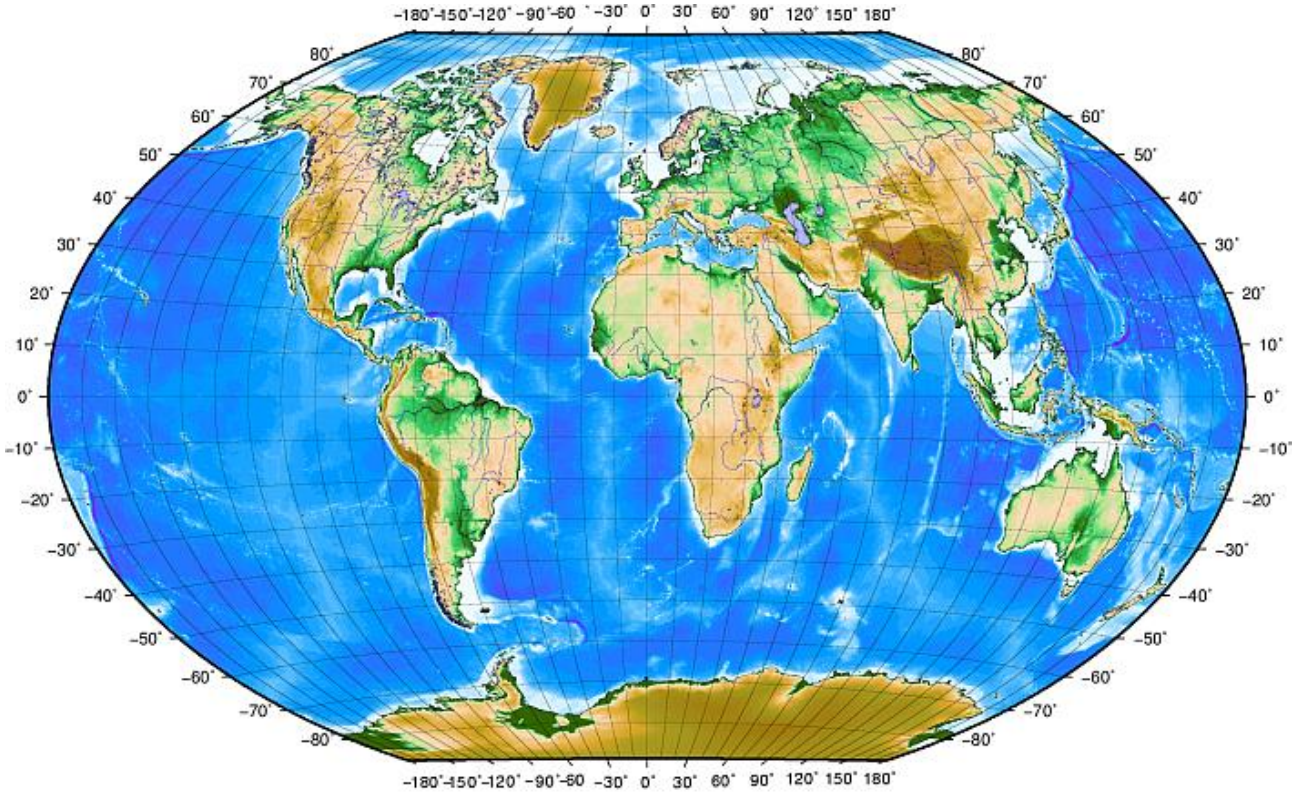
2. importing shapefiles

5.1 Aside – Geographic Information Systems

Projected Coordinate Reference System (CRS) - UTM

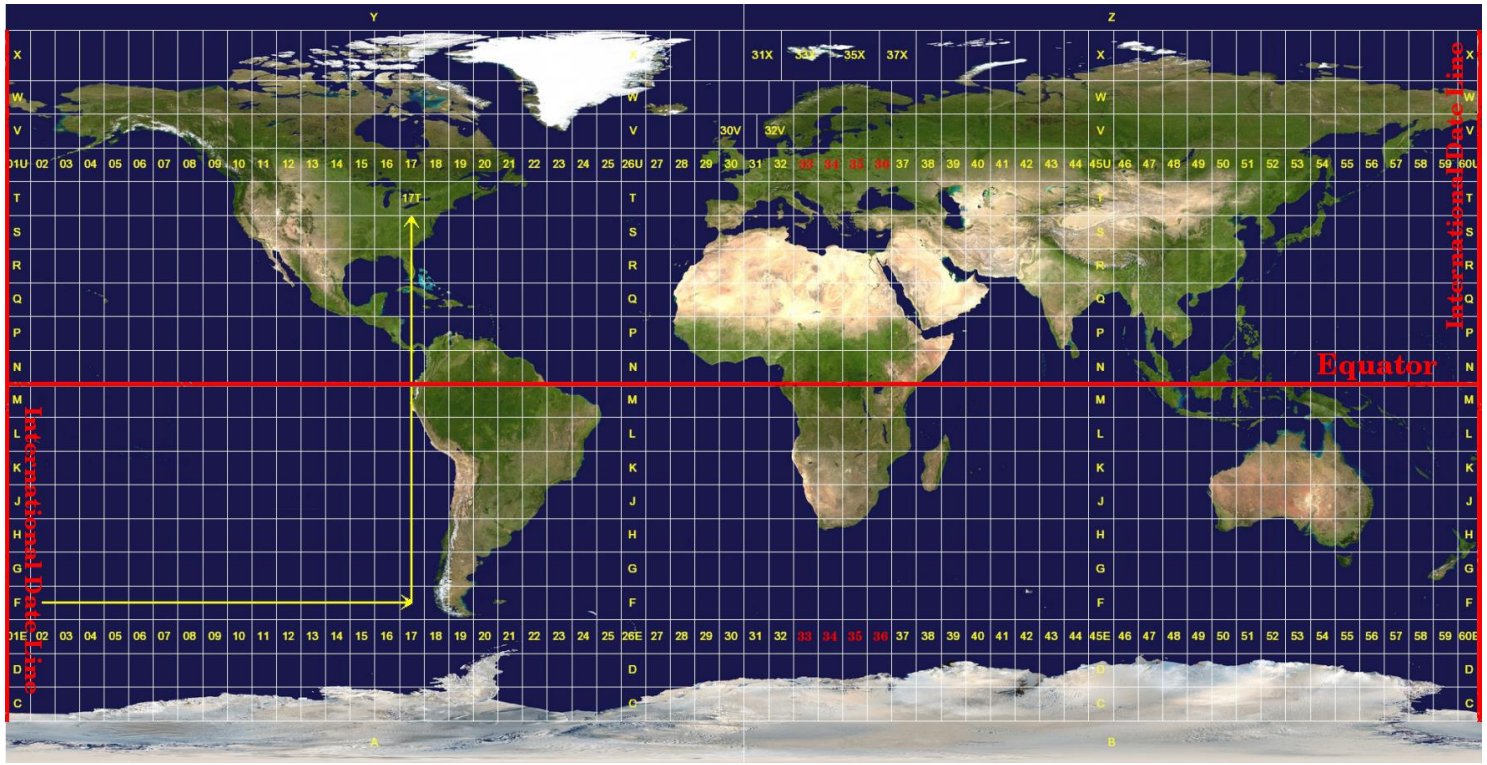
5.1 Aside – Geographic Information Systems

Geographic Coordinate Reference System (CRS) – WGS84



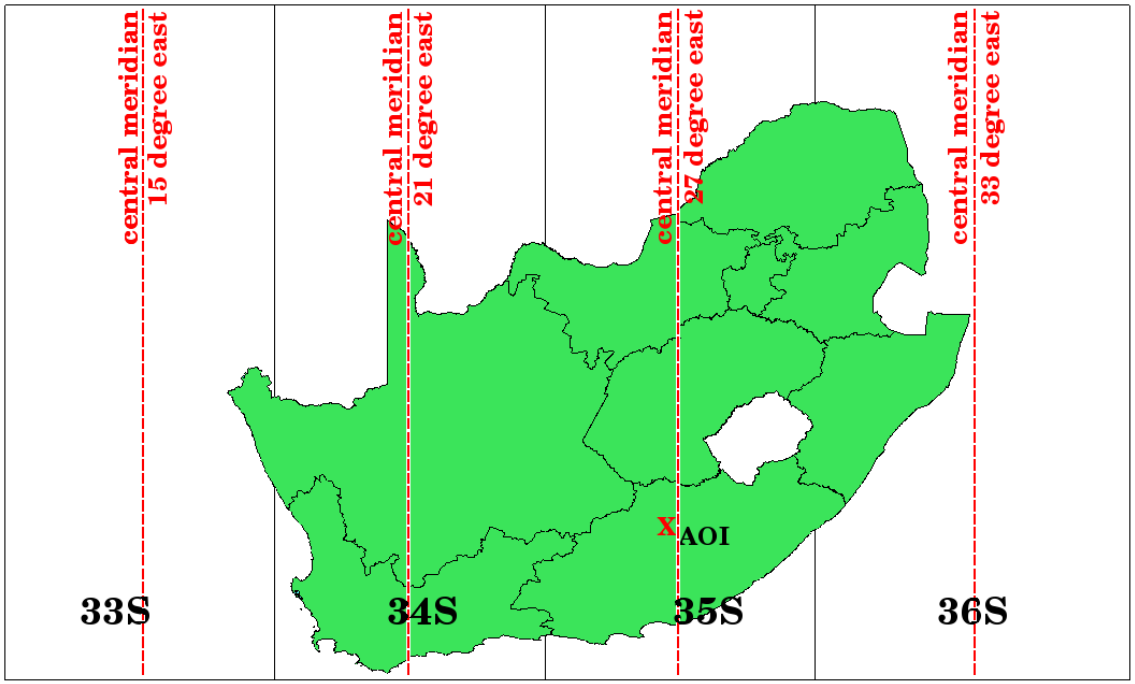
5.1 Aside – Geographic Information Systems

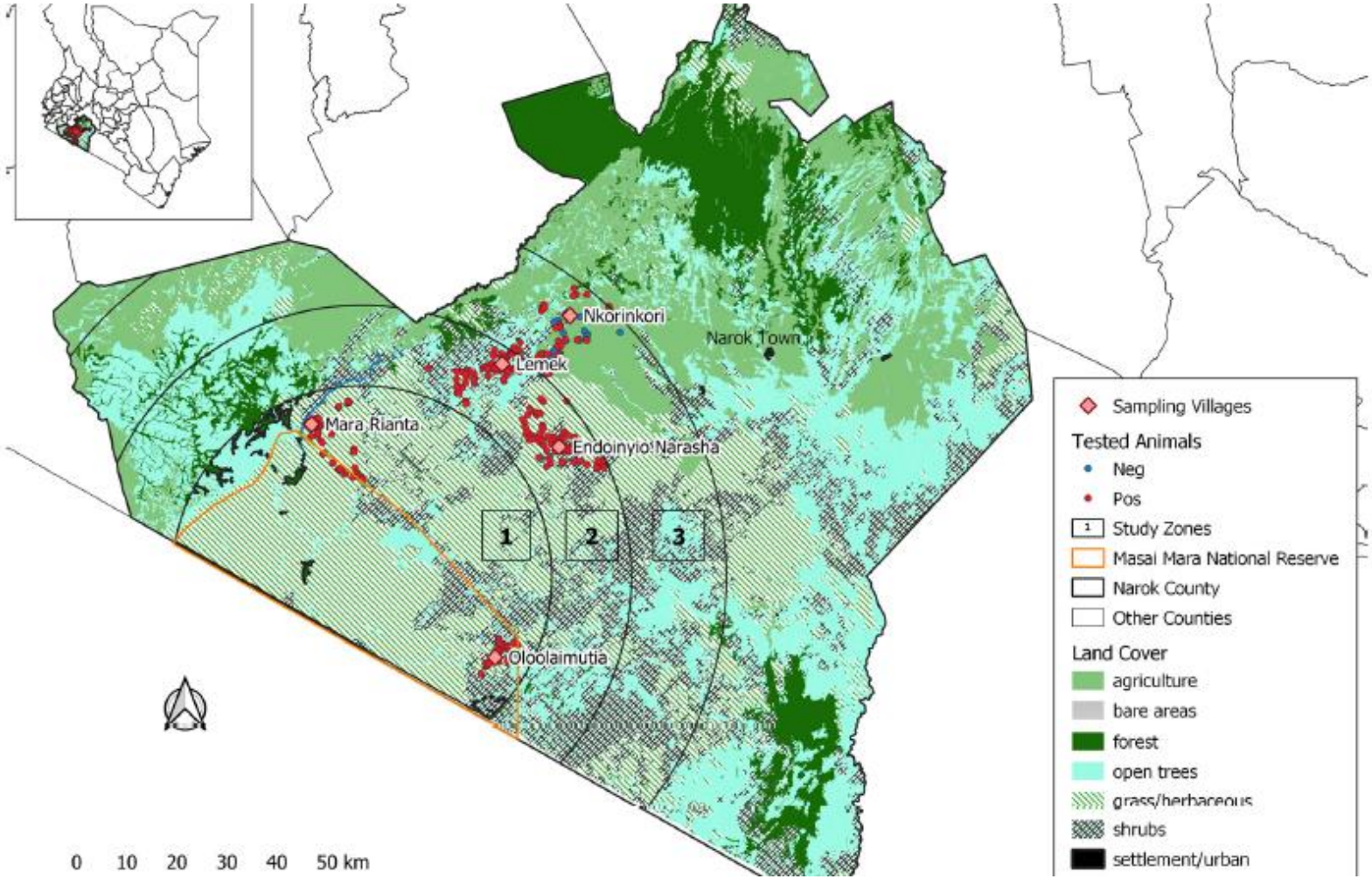
Projected Coordinate Reference System (CRS) - UTM



5.1 Aside – Geographic Information Systems

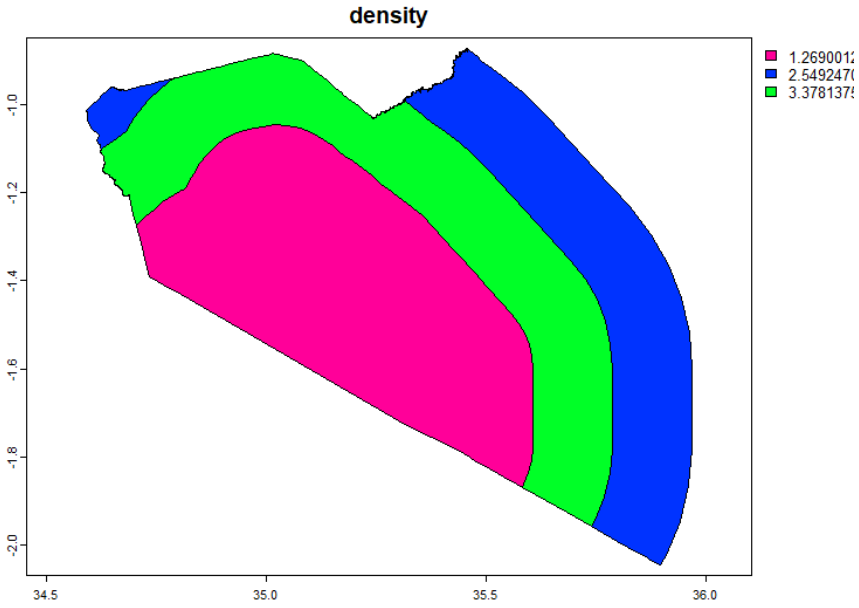
Projected Coordinate Reference System (CRS) – UTM Botswana 35S



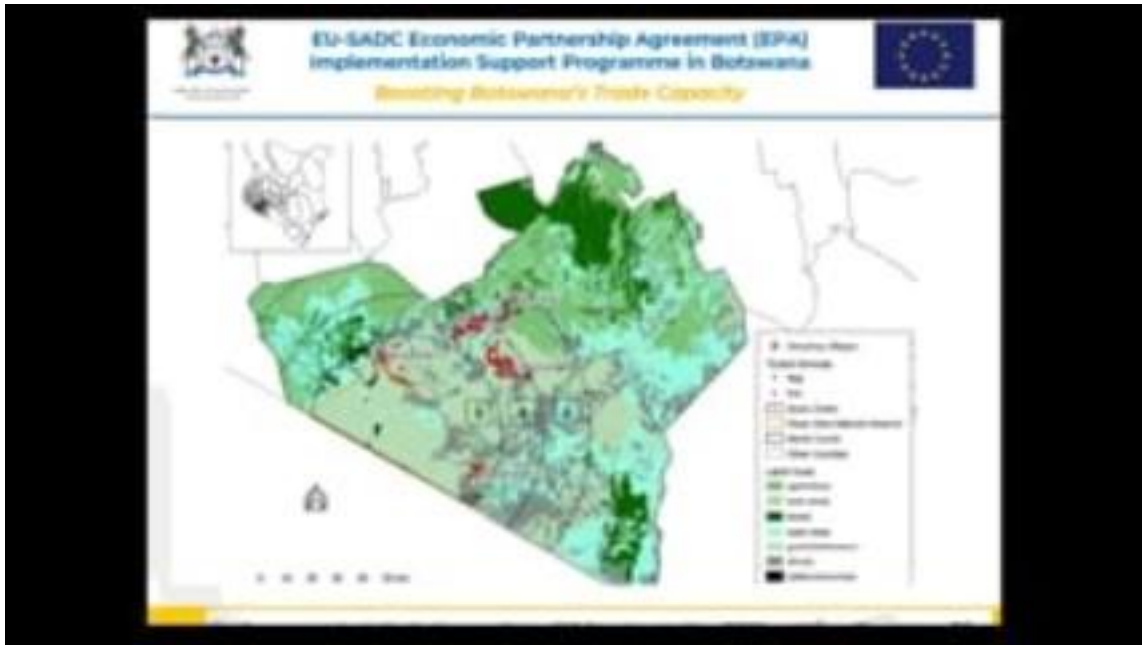

Nkorinkori
Narok Town
Lemek
Mara Rianta
Endoinyio Narasha
1
2
3
Oloolaimutia
0 10 20 30 40 50 km
Sampling Villages
Tested Animals
Neg
Pos
Study Zones
Masai Mara National Reserve
Narok County
Other Counties
Land Cover
agriculture
bare areas
forest
open trees
grass/herbaceous
shrubs
settlement/urban

5.3.3.2.2 Cattle density has been calculated for each buffer region

Go back to the point data `sampLoc_region` and allocate the cattle density to each point



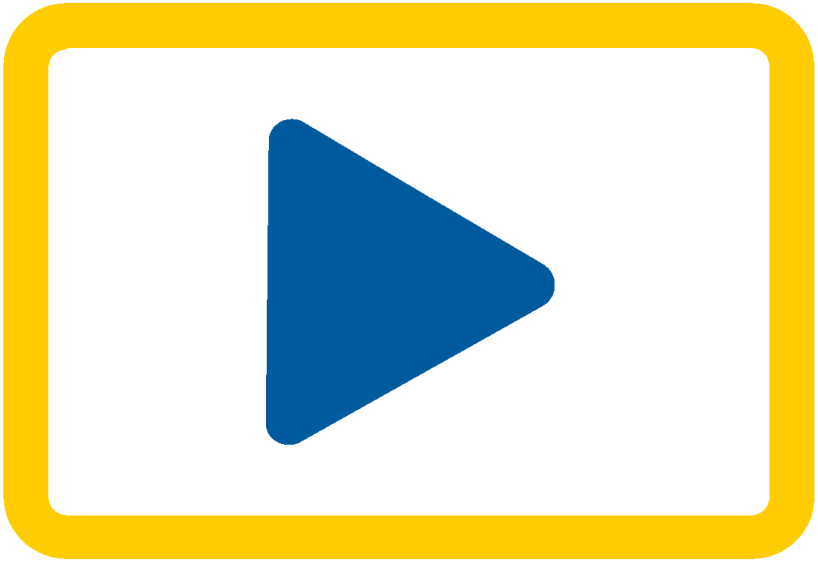
5.6 DEMO - qGIS


EU-SADC Economic Partnership Agreement (EPA)
Implementation Support Programme in Botswana

5.7 – Create the map in qGIS yourselves

OR

Create a similar map of FMD cases in Botswana (see the Shapefiles folder for a Botswana FMD zone layer)



Theme 5: END

Theme 6: Communicating R Analysis using RMarkdown

R Markdown Cheat Sheet

learn more at rmarkdown.rstudio.com

rmarkdown 0.2.50 Updated: 8/14

RStudio

1. Workflow

R Markdown is a format for writing reproducible, dynamic reports with R. Use it to embed R code and results into slideshows, pdfs, html documents, Word files and more. To make a report:

i. Open - Open a file that uses the .Rmd extension.

ii. Write - Write content with the easy to use R Markdown syntax

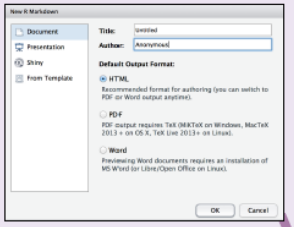
iii. Embed - Embed R code that creates output to include in the report

iv. Render - Replace R code with its output and transform the report into a slideshow, pdf, html or ms Word file.

2. Open File

Start by saving a text file with the extension .Rmd, or open an RStudio Rmd template

- In the menu bar, click **File ▸ New File ▸ R Markdown…**
- A window will open. Select the class of output you would like to make with your .Rmd file
- Select the specific type of output to make with the radio buttons (you can change this later)
- Click OK



3. Markdown

Next, write your report in plain text. Use markdown syntax to describe how to format text in the final report.

syntax

```
Plain text
End a line with two spaces to start a new paragraph.
*italics* and _italics_
**bold** and __bold__
superscript^2^
~~strikethrough~~
[link](www.rstudio.com)

# Header 1

## Header 2

### Header 3

#### Header 4

##### Header 5

###### Header 6

endash: --
emdash: ---
ellipsis: ...
inline equation: $A = \pi*r^{2}$
image: ![](path/to/smallorb.png)

horizontal rule (or slide break):

***

> block quote

* unordered list
* item 2
    + sub-item 1
    + sub-item 2

1. ordered list
2. item 2
    + sub-item 1
    + sub-item 2

Table Header  | Second Header
------------- | -------------
Table Cell    | Cell 2
Cell 3        | Cell 4
```

becomes

Plain text
End a line with two spaces to start a new paragraph.
italics and *italics*
bold and **bold**
superscript^2
~~strikethrough~~
link

Header 1
Header 2
Header 3
Header 4
Header 5
Header 6

endash: –
emdash: —
ellipsis: …
inline equation: $A = \pi * r^2$
image:
horizontal rule (or slide break):

> block quote

- unordered list
- item 2
  - sub-item 1
  - sub-item 2

1. ordered list
2. item 2
   - sub-item 1
   - sub-item 2

Table Header	Second Header
Table Cell	Cell 2
Cell 3	Cell 4

4. Choose Output

Write a YAML header that explains what type of document to build from your R Markdown file.

YAML

A YAML header is a set of key: value pairs at the start of your file. Begin and end the header with a line of three dashes (- - -)

```
---
title: "Untitled"
author: "Anonymous"
output: html_document
---

This is the start of my
report. The above is metadata
saved in a YAML header.
```

The RStudio template writes the YAML header for you

The output value determines which type of file R will build from your .Rmd file (in Step 6)

output: html_document	html file (web page)
output: pdf_document	pdf document
output: word_document	Microsoft Word .docx
output: beamer_presentation	beamer slideshow (pdf)
output: ioslides_presentation	ioslides slideshow (html)

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudio.com

5. Embed Code

Use knitr syntax to embed R code into your report. R will run the code and include the results when you render your report.

inline code

Surround code with back ticks and r.
R replaces inline code with its results.

```
Two plus two equals `r 2 + 2`.
```

Two plus two equals 4.

code chunks

Start a chunk with ```{r}.
End a chunk with ```

```
Here's some code
```{r}
dim(iris)
```
```

Here's some code

```
dim(iris)
```

```
## [1] 150   5
```

display options

Use knitr options to style the output of a chunk.
Place options in brackets above the chunk.

```
Here's some code
```{r eval=FALSE}
dim(iris)
```
```

Here's some code

```
dim(iris)
```

```
Here's some code
```{r echo=FALSE}
dim(iris)
```
```

Here's some code

```
## [1] 150   5
```

option	default	effect
eval	TRUE	Whether to evaluate the code and include its results
echo	TRUE	Whether to display code along with its results
warning	TRUE	Whether to display warnings
error	FALSE	Whether to display errors
message	TRUE	Whether to display messages
tidy	FALSE	Whether to reformat code in a tidy way when displaying it
results	"markup"	"markup", "asis", "hold", or "hide"
cache	FALSE	Whether to cache results for future renders
comment	"##"	Comment character to preface results with
fig.width	7	Width in inches for plots created in chunk
fig.height	7	Height in inches for plots created in chunk

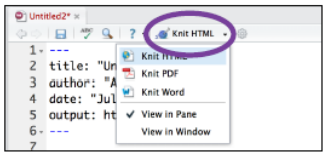
For more details visit yihui.name/knitr/

6. Render

Use your .Rmd file as a blueprint to build a finished report.

Render your report in one of two ways

1. Run **rmarkdown::render("<file path>")**
2. Click the **knit HTML** button at the top of the RStudio scripts pane



When you render, R will

- execute each embedded code chunk and insert the results into your report
- build a new version of your report in the output file type
- open a preview of the output file in the viewer pane
- save the output file in your working directory

7. Interactive Docs

Turn your report into an interactive Shiny document in 3 steps

1. Add **runtime: shiny** to the YAML header

```
---
title: "Line graph"
output: html_document
runtime: shiny
---
```

2. In the code chunks, add Shiny **input** functions to embed widgets. Add Shiny **render** functions to embed reactive output

```
---
title: "Line graph"
output: html_document
runtime: shiny
---

Choose a time series:
```{r echo = FALSE}
selectInput("data", "",
  c("co2", "lh"))
```

See a plot:
```{r echo = FALSE}
renderPlot({
  d <- get(input$data)
  plot(d)
})
```
```

3. Render with **rmarkdown::run** or click **Run Document** in RStudio

* *Note: your report will be a Shiny app, which means you must choose an html output format, like* ***html_document*** *(for an interactive report) or* ***ioslides_presentation*** *(for an interactive slideshow).*

8. Publish

Share your report where users can visit it online

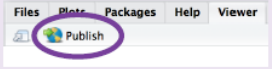
Rpubs.com

Share non-interactive documents on RStudio's free R Markdown publishing site
www.rpubs.com

ShinyApps.io

Host an interactive document on RStudio's server. Free and paid options
www.shinyapps.io

Click the "Publish" button in the RStudio preview window to publish to rpubs.com with one click.



9. Learn More

Documentation and examples - rmarkdown.rstudio.com
Further Articles - shiny.rstudio.com/articles
- blog.rstudio.com
- @rstudio

RStudio

RStudio® and Shiny™ are trademarks of RStudio, Inc.
CC BY RStudio info@rstudio.com
844-448-1212 rstudio.com

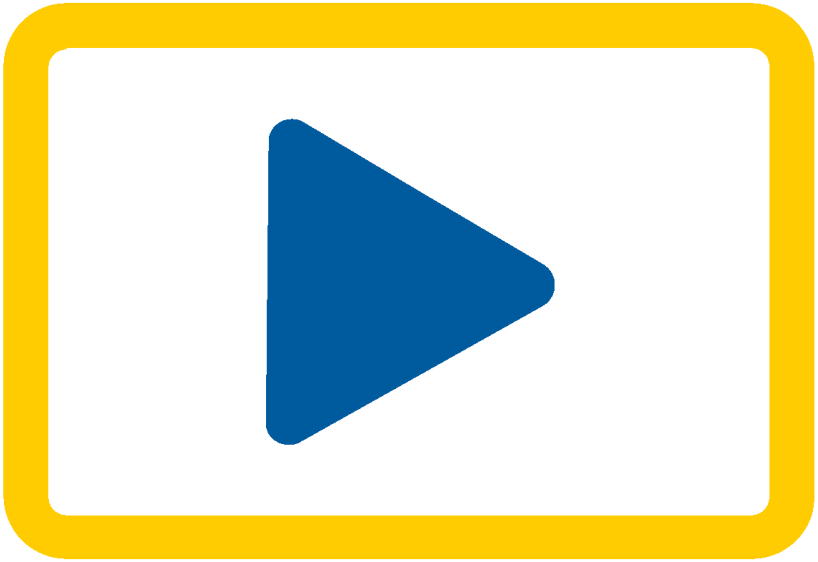
Open 06_RMarkdown.Rmd

Open 06_Analysis.Rmd

Use the existing Rmarkdown document and add a section from any of the other analysis we have done based on the Nthiwa et al. manuscript

OR

Use one of your own datasets and make a basic start to a report



Theme 6: END

Theme 7: Further analysis of diagnostic tests

Diagnostic test evaluation is a complex topic.

- Typical issues with non-gold standards
- Where gold standards exist then analysis is manageable and pretty much descriptive as sensitivity, specificity and other test characteristics
- We have looked at interpreting diagnostic tests in parallel and an series
- To further our knowledge we will consider the
  - Kappa statistic
  - Descriptive stats on Ct values

KAPPA statistic

- Kappa is a measure of agreement beyond the level of agreement expected by chance alone. The observed agreement is the proportion of samples for which both methods (or observers) agree (see ?epiR::epi.kappa)
- Can be used to evaluate an opinion as well as a diagnostic test
- Flip coin by two different people and establish Kappa

Open 07_Tests.R

Descriptive stats on Ct values

- Source of Ct based testing
- Discuss

Theme 7: END

Self projects

1. Evaluate the WAHID FMD data. Make a Rmarkdown document that provides descriptive statistics of the FMD reporting including an Epidemic curve
2. Using the *botswanaFMDControl* shapefile establish the average Cattle density within each region in the country
    1. Using the case data establish the average of the average prevalence for each overarching outbreak within each region and plot this as a choropleth map
3. Create a publishable map in qGIS indicating the underlying cattle density, the FMD control zones, the locations of outbreaks that have been reported to the OIE

References and Sources

- https://www.sciencedirect.com/science/article/pii/S0167587719305380?via%3Dihub
  - https://data.mendeley.com/datasets/8drcnw9y2k/1
- Spatial – naturalearthdata.com
- OIE WAHID (www.oie.int)
- GLW: Gilbert M, G Nicolas, G Cinardi, S Vanwambeke, TP Van Boeckel, GRW Wint, TP Robinson (2018) Global Distribution Data for Cattle, Buffaloes, Horses, Sheep, Goats, Pigs, Chickens and Ducks in 2010. Nature Scientific data, 5:180227. doi: 10.1038/sdata.2018.227
  - http://geoportal.icpac.net/

- epiR Tools for the Analysis of Epidemiological Data – 2022
- Charles DiMaggio – an introduction to R for epidemiologists
- https://rspatial.org
- https://happygitwithr.com/

Thank you

Ke a Leboga