

**An Introduction to**

# **GRASS**

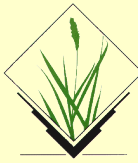
The

## **GRASS Story**

*<http://grass.osgeo.org>*

Markus Metz, Markus Neteler

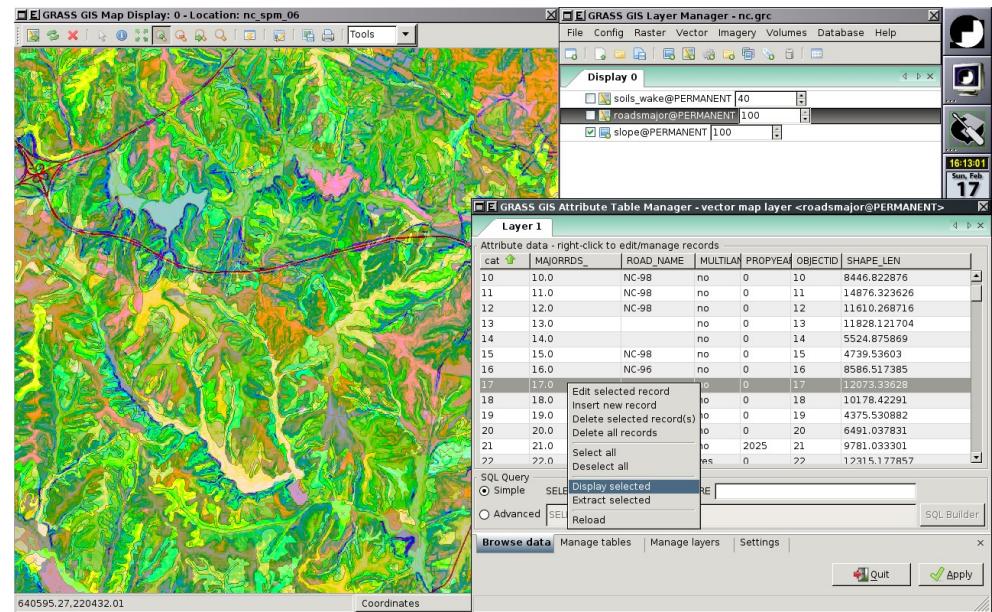
# Structure of this seminar



Introduction to GRASS GIS

Impart working knowledge of GRASS

Exercises on statistical procedures in GRASS





# What is GRASS GIS ?

**Geographic Resources AnalYSIS Support System**

Free and Open Source since 1984

Member of  **OSGeo**  
*Your Open Source Compass*

Linked to GDAL,  GIS and 

Portable: GNU/Linux, Mac OSX, MS-Windows, SUN, etc.



# GRASS Features

Raster 2D/3D processing

Vector 2D/3D processing

Native raster and vector format

3D visualization system

Database Management System (DBMS)

DBF, sqlite, PostgreSQL, MySQL

# GRASS database concept



GRASS database

folder with Locations

Location

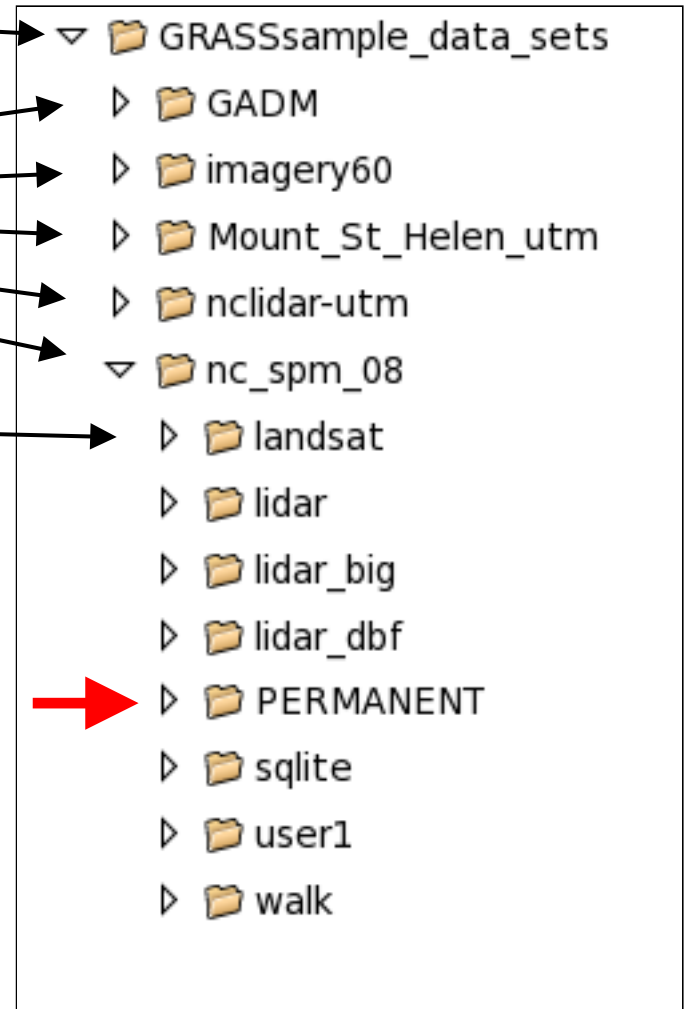
folder with Mapsets

Mapset

collection of maps and  
support data

Each Location

- is defined by a projection
- can hold several mapsets



# GRASS database concept



## ***Exercise***

start GRASS

any way you like, but with a GUI please

# GRASS database concept

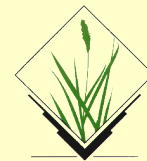


## ***Exercise***

open the North Carolina dataset

```
> grass64 -wx /home/user/GRASS_sample_datasets/nc_spm_08/geostat2010/
```

# GRASS



## Exerc

open

```
> grass  
spm_0
```

ets/nc\_



# GRASS database concept



## ***Exercise***

get projection information for the North Carolina  
Location

command: `g.proj`

wxGUI: Config -> Manage projections -> Manage projections

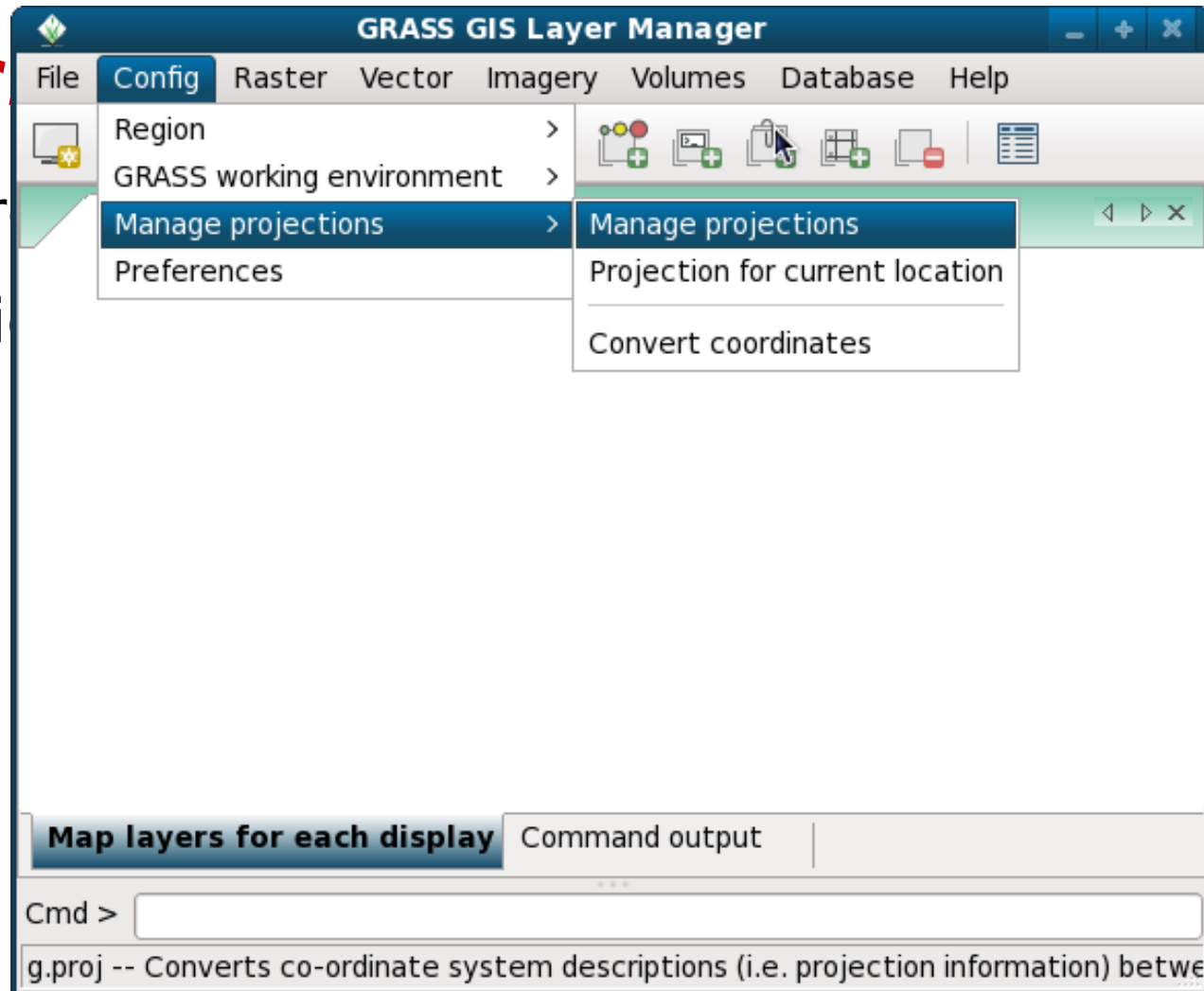
# GRASS database concept



**Exerc**

get pr

Locati



na

# GRASS database concept



## *Exercise*

get projection information for the North Carolina  
Location

```
> g.proj -p  
> g.proj -j  
> g.proj -w  
> g.proj -e
```

# GRASS database concept



## *Exercise*

```
name      : Lambert Conformal Conic
proj      : lcc
datum     : nad83
a         : 6378137.0
es        : 0.006694380022900787
lat_1     : 36.166666666666666
lat_2     : 34.333333333333334
lat_0     : 33.75
lon_0     : -79
x_0       : 609601.22
y_0       : 0
units     : meters
```

```
> g.proj -p
> g.proj -j
> g.proj -w
> g.proj -e
```

# GRASS database concept



## *Exercise*

```
name      : Lambert Conformal Conic
proj      : lcc
datum     : nad83
a         : 6378137.0
es        : 0.006694380022900787 } ellipsoid GRS80
lat_1     : 36.166666666666666   } standard parallels
lat_2     : 34.333333333333334
lat_0     : 33.75                reference latitude
lon_0     : -79                  reference longitude
x_0       : 609601.22            } lon and lat shifts
y_0       : 0
units     : meters
```

```
> g.proj -p
> g.proj -j
> g.proj -w
> g.proj -e
```

# GRASS database concept



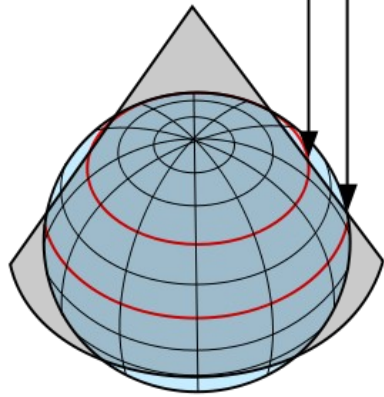
## ***Exercise***

projection of the North Carolina Location:

Identical to NAD83(HARN) / North Carolina

EPSG code: 3358

Two standard parallels  
(selected by mapmaker)



# GRASS database concept



Locations are defined by projection

→ transfer maps between locations = map reprojection

Raster reprojection

set desired extends and resolution prior to reprojection

Vector reprojection

the whole vector map is reprojected by coordinate conversion



# GRASS command structure

<b><i>prefix</i></b>	<b><i>function class</i></b>	<b><i>type of command</i></b>	<b><i>example</i></b>
g.*	general	general data management	<i>g.rename: renames map</i>
d.*	display	graphical output	<i>d.rast: display raster map</i> <i>d.vect: display vector map</i>
r.*	raster	raster processing	<i>r.mapcalc: map algebra</i> <i>r.univar: univariate statistics</i>
v.*	vector	vector processing	<i>v.clean: topological cleaning</i>
i.*	imagery	imagery processing	<i>i.pca: Principal Components Analysis on imagery group</i>
r3.*	voxel	3D raster processing	<i>r3.stats: Voxel statistics</i>
db.*	database	database management	<i>db.select: select value(s) from table</i>
ps.*	postscript	map creation in PostScript format	<i>ps.map: PostScript map creation</i>

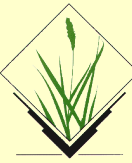
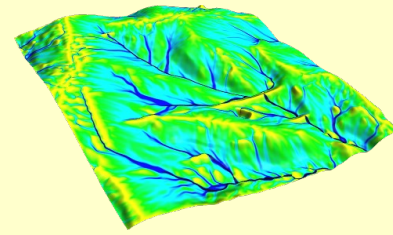


# **Raster intro**

computational region

raster import / export via GDAL

# Raster intro



## Computational region

- defined by extends and resolution
- applies to raster operations

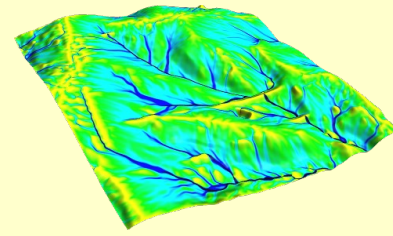
## Raster region

- defined by extends and resolution
- each raster map has its own region
- computational region overrides raster region

## Display region

- extends of the current map display
- independent of the current computational region and the raster region

# Raster intro

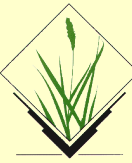
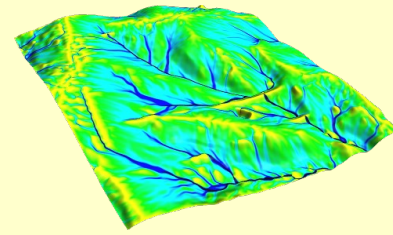


## Computational region

command: `g.region`

wxGUI: Config -> Region -> Set region

# Raster intro



## NULL values and MASK

NULL values: e.g. gaps in DEM

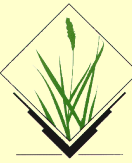
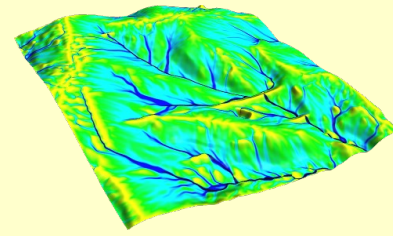
MASK: mask out areas outside region of interest / study area

MASK'ed cells are read as NULL cells -> usually skipped

## Raster import/export via GDAL

105 supported formats in gdal version 1.7.1

# Raster intro

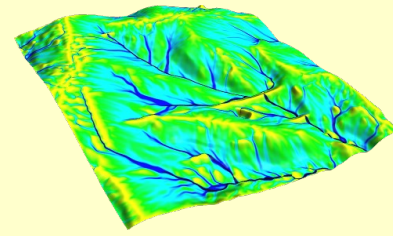


## ***Exercise***

load raster map to display

zoom into map

# Raster intro



## *Exercise*

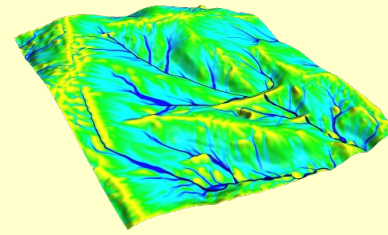
load raster map to display, check region

```
> g.region -p
```

zoom into map, check region

```
> g.region -p
```

# Raster intro



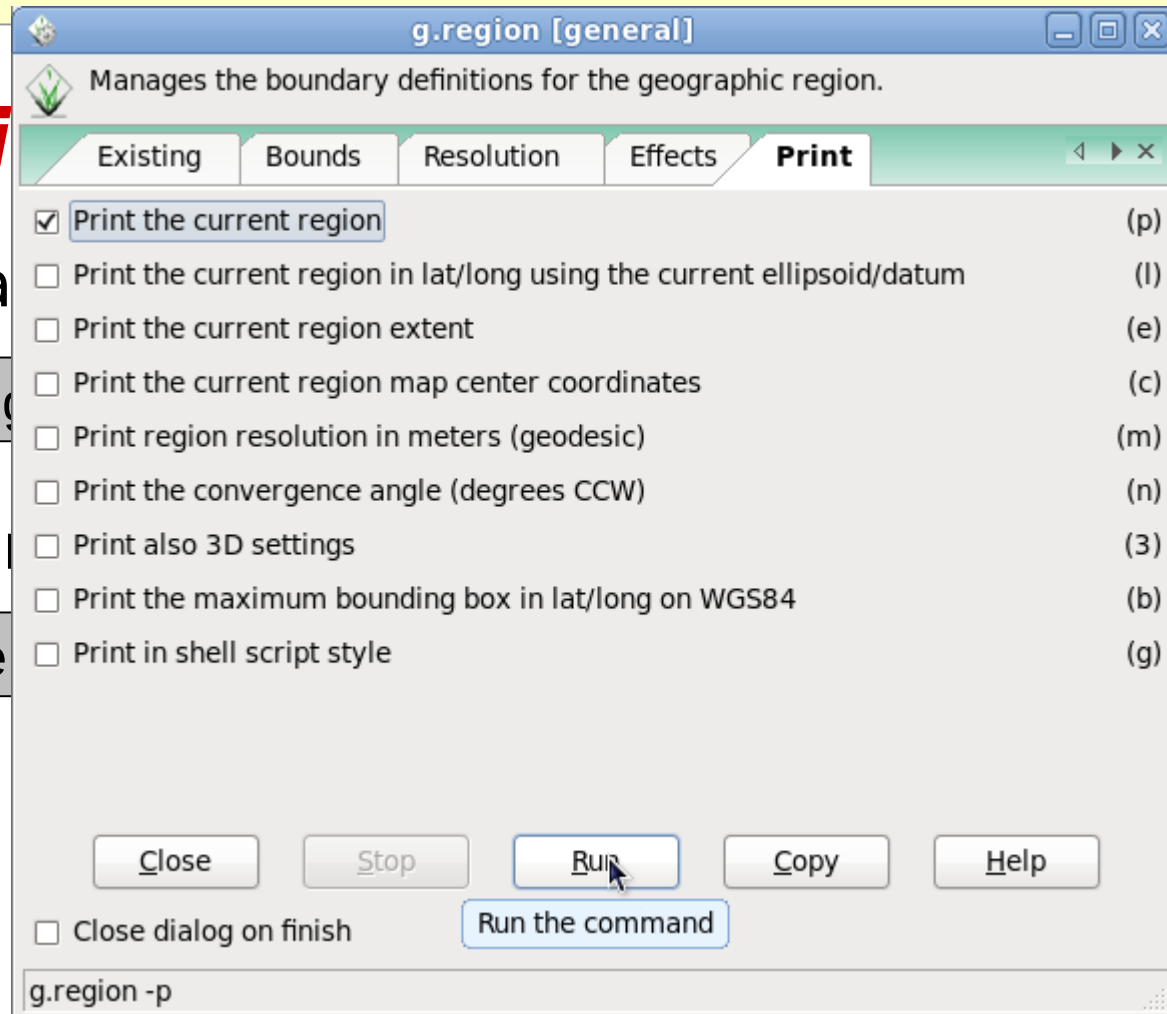
## Exerci

load ra

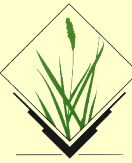
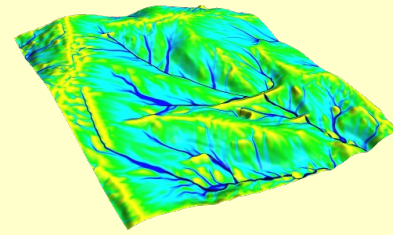
```
> g.region
```

zoom in

```
> g.region
```



# Raster processing



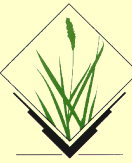
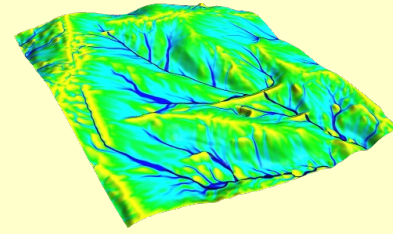
## Import/export raster maps

`r.in.*` and `r.out.*`

`r.in.gdal` always imports the complete map

**!! Raster export adheres to computational region !!**

# Raster processing



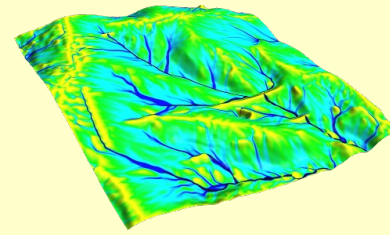
## ***Exercise***

Raster export via GDAL

command: `r.out.gdal`

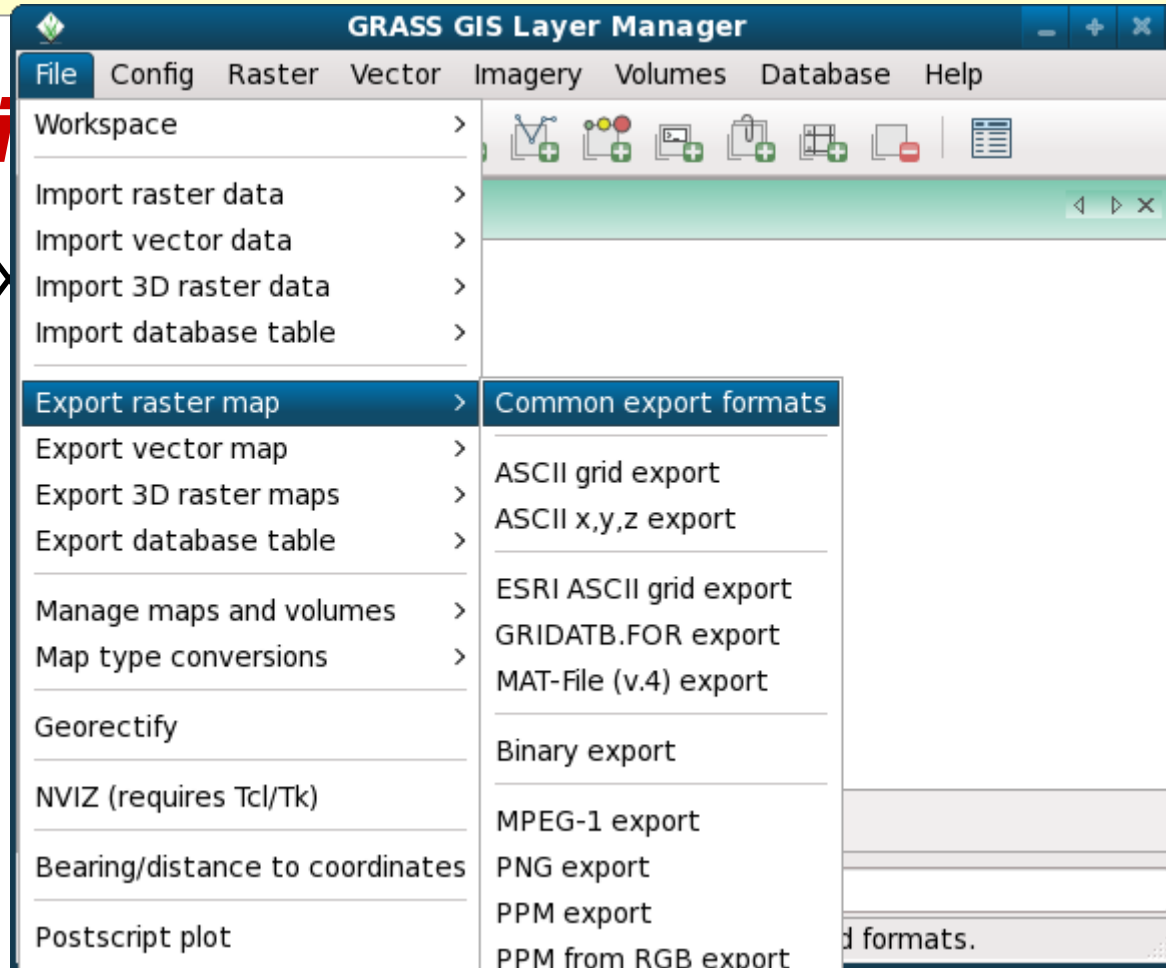
wxGUI: File -> Export raster map -> Common export formats

# Raster processing

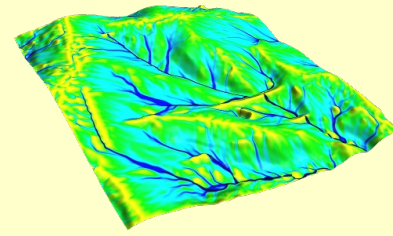


**Exerci**

Raster ex



# Raster processing



## *Exercise*

### Raster export via GDAL

set region to some raster map

```
> g.region -p rast=<raster>
```

export this raster with r.out.gdal

```
> r.out.gdal input=<raster> output=<raster>.tif
```

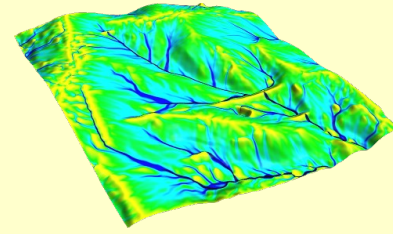
display this raster, zoom in, set region from display

export again with r.out.gdal

compare size of the two exported raster maps

compare output of gdalinfo

# Raster processing



## Example raster module groups

- resampling

- Reprojection/georectification

- map calculator

- Terrain analysis

- Hydrologic modeling

- Reports and statistics

Raster maps: DEMs, land cover, climatic maps ...

Imagery maps: Landsat, MODIS, SPOT, QuickBird ...

The background of the slide is a 3D visualization of a city model. It shows a dense urban area with numerous yellow and orange rectangular blocks representing buildings. Some buildings are colored blue or green. The city is situated in a valley, with green hills and a river visible in the upper left. A semi-transparent white rectangular box is overlaid on the center of the image, containing the title and a list of features.

# Vector intro

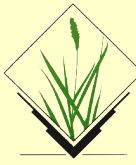
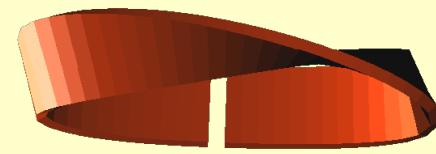
native vector format

vector import/export via OGR

vector topology

vector 2D/3D support

# Vector intro



## Native vector format

Vector topology

m:n mapping of geometry features to attributes

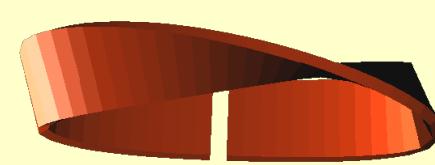
Vector layers

OGC Simple Features <-> Topological Vector Conversion

Database Management system (DBMS)

SQLite, PostgreSQL + PostGIS, MySQL(, DBF)

# GRASS Vector model



## Vector geometry types

Point

Centroid

Line

Boundary

Area (Boundaries + Centroid)

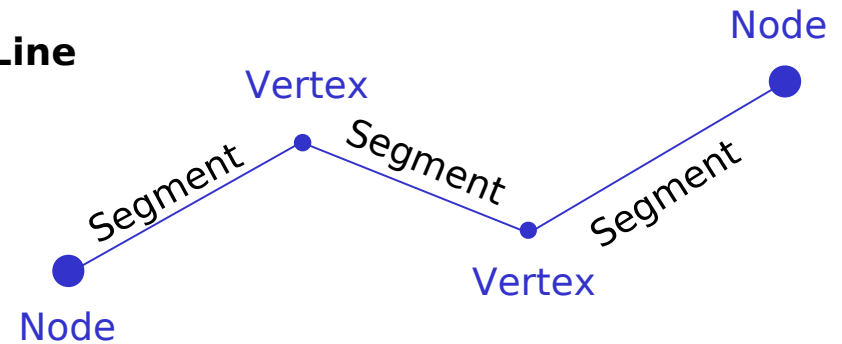
Face (3D Area)

[Kernel (3D Centroid)]

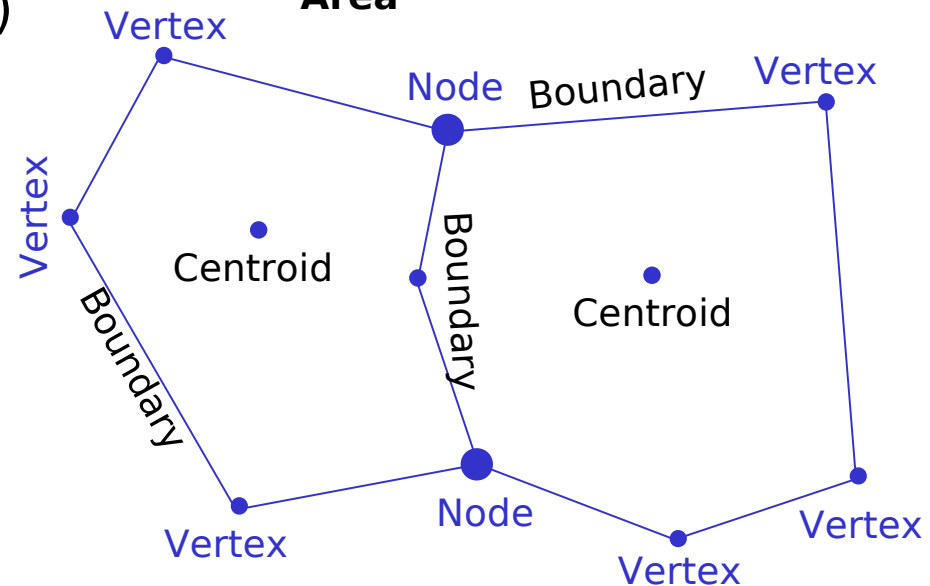
[Volumes (Faces + Kernel)]

All types are **true 3D**: x,y,z

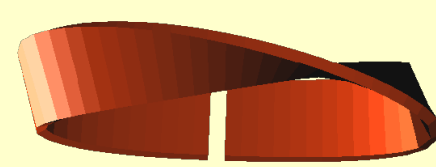
Line



Area



# Vector processing



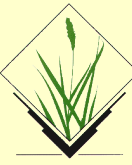
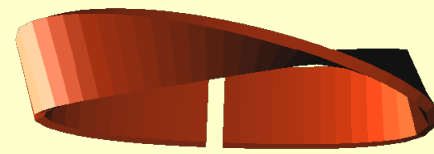
## Import/export vector maps

v.in.\* and v.out.\*

v.out.ogr always exports the whole map

OGR supported formats: 40

# Vector processing



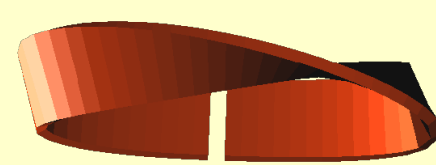
## ***Exercise***

Vector import via OGR

command: `v.in.ogr`

wxGUI: File -> Import vector data -> Import vector data

# Vector processing



**Exercise**  
Vector

**v.in.ogr [vector, import]** [minimize] [maximize] [close]

Convert OGR vector layers to GRASS vector map.

**Required** Selection Subregion Min-area & snap Attributes [left] [right] [close]

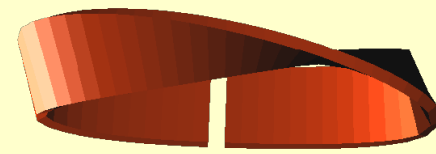
OGR datasource name: (dsn, string)

Name for output vector map: (output, string)  
 [dropdown arrow]

☐ Add created map into layer tree  
☐ Close dialog on finish

v.in.ogr dsn=ncshape/boundary\_county.shp output=boundary\_county

# Vector processing



## ***Exercise***

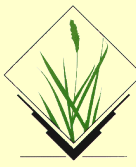
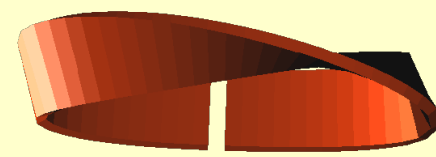
Vector import via OGR

Read output:

Cleaning procedures

Summary of topology building

# Vector processing



## ***Exercise***

### Vector import via OGR

Cleaning procedures for area import

- break polygons

- remove duplicates

- break boundaries

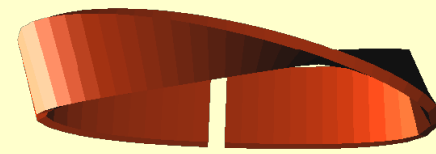
- remove duplicates

- clean boundaries at nodes

- change dangles to lines

- remove bridges

# Vector processing



## *Exercise*

### Vector import via OGR

#### Summary of topology building

Number of nodes: 2040

Number of primitives: 2836

Number of points: 0

Number of lines: 0

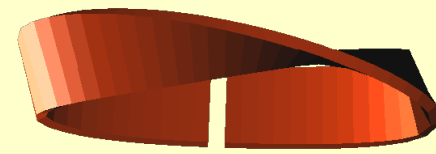
Number of boundaries: 1910

Number of centroids: 926

Number of areas: 926

Number of isles: 130

# Vector processing



## Summary of topology building

No need to worry about

Number of areas without centroid:

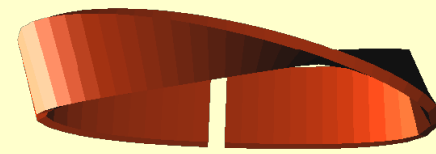
... but watch out for

Number of incorrect boundaries:

Number of centroids outside area:

Number of duplicate centroids:

# Vector processing



## Example vector module groups

Topological geometry feature editing

LiDAR analysis

Linear referencing

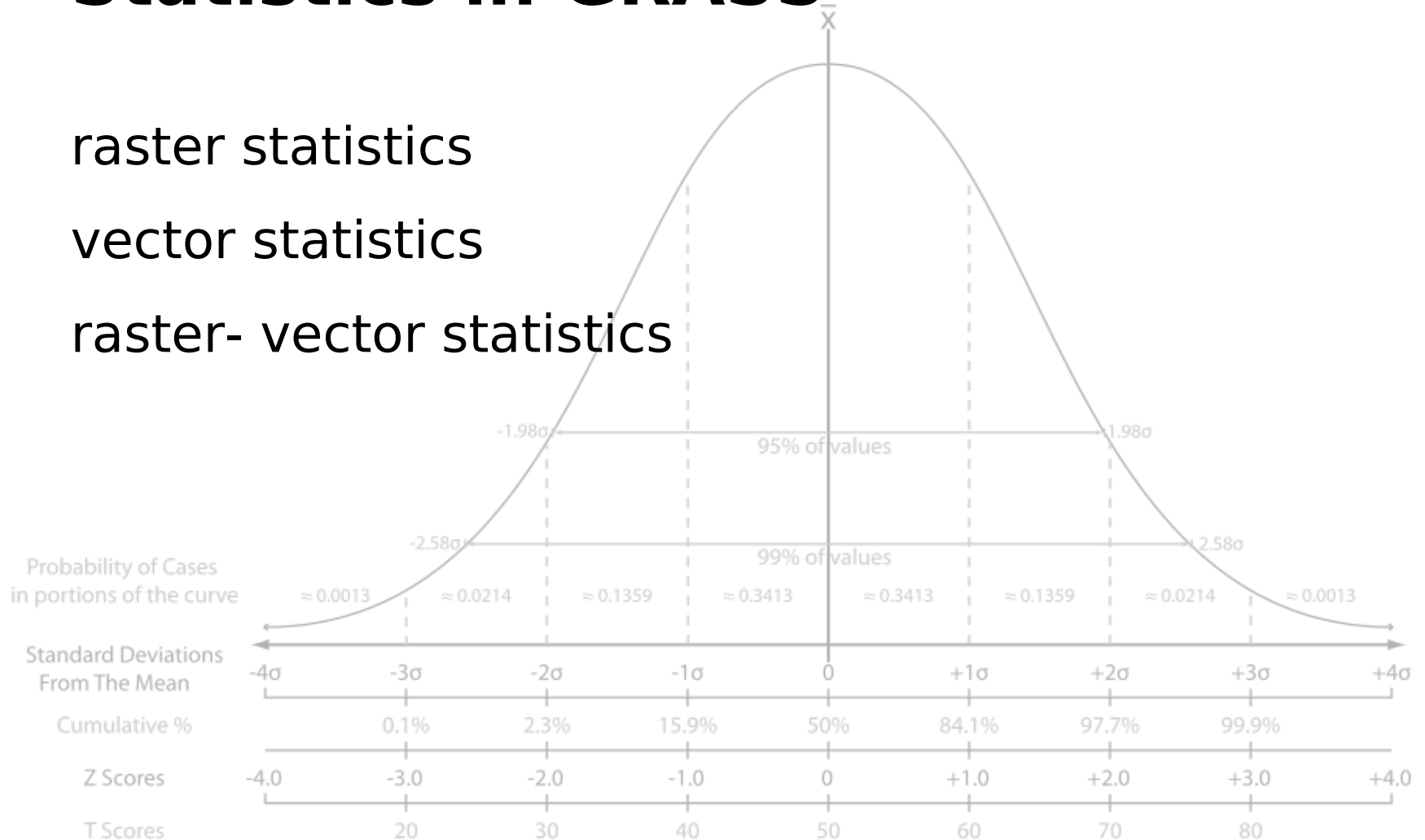
Network analysis

# Statistics in GRASS

raster statistics

vector statistics

raster- vector statistics



# Statistical modules



## Statistics on one raster

### General statistics

- General area statistics

- Range

- Quantiles

- Sum per category

- Univariate statistics

- Sample transects

### Neighbourhood analysis

### Resampling using aggregate statistics

# Statistical modules



## General statistics

GUI: Raster -> Reports and statistics

# Statistical modules



## General area statistics

command: `r.stats`

wxGUI: Raster -> Reports and statistics -> General statistics

# Statistical modules



## General area statistics

```
> r.stats -al input=landclass96@PERMANENT
```

# Statistical modules



General area

> r.stats

**r.stats [raster, statistics]**

Generates area statistics for raster map layers.

**Required** **Print** Optional Command output

- ☐ Print averaged values instead of intervals (A)
- ☒ Print area totals (a)
- ☐ Print cell counts (c)
- ☐ Print APPROXIMATE percents (total percent may not be 100%) (p)
- ☒ Print category labels (l)
- ☐ Print grid coordinates (east and north) (g)
- ☐ Print x and y (column and row) (x)
- ☐ Print raw indexes of fp ranges (fp maps only) (r)

Close Stop Run Copy Help

☐ Close dialog on finish

r.stats -a -l input=landclass96@PERMANENT

ENT

# Statistical modules



## Range

command: `r.describe`

wxGUI: Raster -> Reports and statistics -> Range of category values

# Statistical modules



## Range

r.describe

```
> g.region -p rast=elev_lid972_1m@PERMANENT
> r.describe map=landclass96@PERMANENT
> r.describe -d map=landclass96@PERMANENT
> g.region -p rast=landclass96@PERMANENT
> r.describe -d map=landclass96@PERMANENT
```

# Statistical modules



Range

r.describe

```
> g.reg
```

```
> r.des
```

```
> r.des
```

```
> g.reg
```

```
> r.des
```

**r.describe [raster]**

Prints terse list of category values found in a raster map layer.

Required Optional Command output Manual

☐ Print the output one value per line (1)

☐ Only print the range of the data (r)

☐ Suppress reporting of any NULLs (n)

☒ Use the current region (d)

☐ Read fp map as integer (i)

☐ Run quietly (q)

☐ Verbose module output (verbose)

☐ Quiet module output (quiet)

String representing no data cell value: (nv, string)

\*

Number of quantization steps: (nsteps, integer)

255

Close Stop Run Copy Help

☐ Close dialog on finish

r.describe -d map=landclass96@PERMANENT

# Statistical modules



## Quantiles

command: `r.quantile`

wxGUI: Raster -> Reports and statistics -> Quantiles for large datasets

# Statistical modules



## Quantiles

r.quantile

```
> g.region -p rast=elev_lid972_1m@PERMANENT  
> r.quantile input=elev_lid972_1m@PERMANENT
```

# Statistical modules



## Sum per category

command: `r.report` (also `r.stats`)

wxGUI: Raster -> Reports and statistics -> Sum area by raster map and category

# Statistical modules



Sum per category

r.report

```
> g.region -p rast=landclass96@PERMANENT  
> r.report map=landclass96@PERMANENT units=h,c,p
```

# Statistics

Sum per cell

r.report

```
> g.region
```

```
> r.report
```

**r.report [raster, statistics]**

Reports statistics for raster map layers.

Required   Formatting   **Optional**

☐ Quiet (q)

☒ Filter out all no data cells (n)

☐ Filter out cells where all maps have no data (N)

☐ Report for cats fp ranges (fp maps only) (C)

☐ Read fp map as integer (use map's quant rules) (i)

☐ Verbose module output (verbose)

☐ Quiet module output (quiet)

Units:

☐ miles

☐ meters

☐ kilometers

☐ acres

☒ hectares

☒ cell counts

☒ percent cover

Name of an output file to hold the report: (output, string)

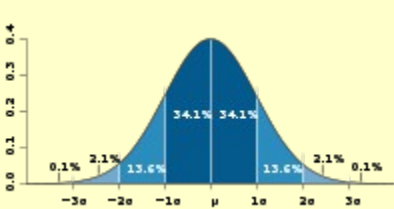
☐ Close dialog on finish

r.report -n map=landclass96@PERMANENT units=h,c,p



units=h,c,p

# Statistical modules



Category Information				cell	%
#	description	hectares	count	cover	
1	developed. . . . .	5935.679	73077	29.31	
2	agriculture. . . . .	173.578	2137	0.86	
3	herbaceous . . . . .	2072.700	25518	10.23	
4	shrubland. . . . .	1366.936	16829	6.75	
5	forest . . . . .	10,255.875	126265	50.64	
6	water. . . . .	430.736	5303	2.13	
7	sediment . . . . .	15.758	194	0.08	
TOTAL		20,251.261	249323	100.00	

# Statistical modules



## Statistics on several rasters

Correlation

Regression

Mutual occurrence

Zonal statistics

Raster map series

# Statistical modules



## Correlation/Covariance

command: `r.covar`

wxGUI: Raster -> Reports and statistics ->

Covariance/correlation

# Statistical modules



## Correlation/Covariance

r.covar

```
> g.region -p rast=elev_lid972_1m@PERMANENT  
> r.covar -r  
map=elevation@PERMANENT,elev_lid792_1m@PERMANENT
```

# Statistical modules



## Regression

command: `r.regression.line`

wxGUI: Raster -> Reports and statistics -> Linear regression

# Statistical modules



## Regression

r.regression.line

```
> g.region -p rast=elevation@PERMANENT  
> r.regression.line -g -s map1=elevation@PERMANENT  
map2=elev_ned_30m@PERMANENT
```

# Statistical modules



## Mutual occurrence

command: `r.coin`

wxGUI: Raster -> Reports and statistics -> Mutual category  
occurrences

# Statistical modules

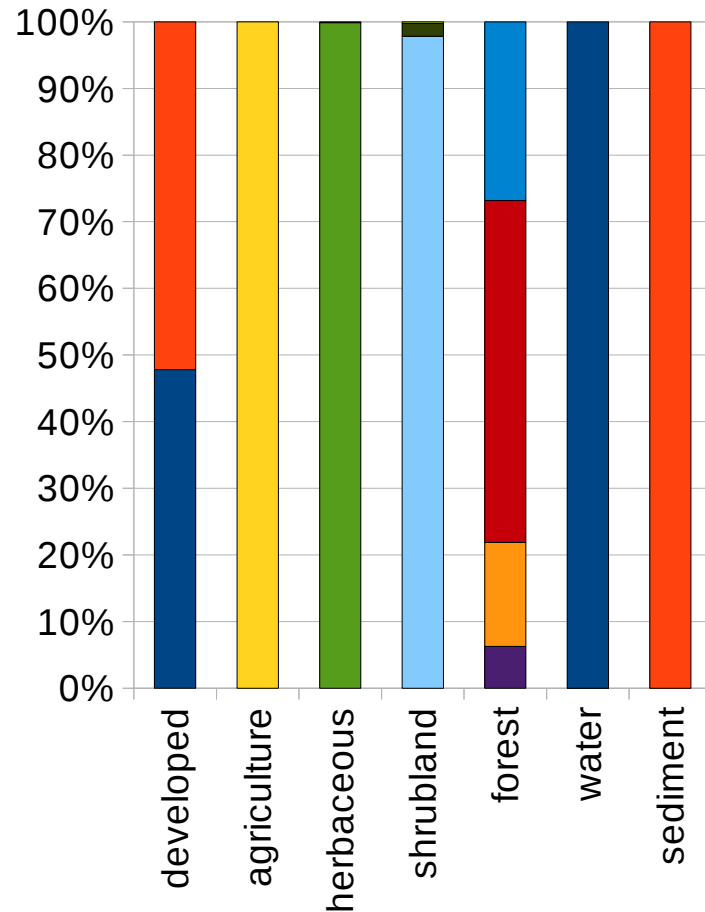


## Mutual occurrence

r.coin

```
> g.region -p rast=landclass96@PERMANENT  
> r.coin map1=landclass96@PERMANENT  
map2=landuse96_28m@PERMANENT units=c
```

# Statistical modules



- Unconsolidated Sediment
- Water Bodies
- Mixed Hardwoods/Conifers
- Southern Yellow Pine
- Bottomland Hardwoods/Hardwood Swamps
- Mixed Hardwoods
- Mixed Shrubland
- Deciduous Shrubland
- Evergreen Shrubland
- Riverine/Estuarine Herbaceous
- Managed Herbaceous
- Cultivated
- Low Intensity Developed
- High Intensity Developed

# Statistical modules



Zonal statistics

command: `r.statistics`, `r.univar.zonal` (addons)

wxGUI: Raster -> Overlay rasters -> Statistical overlay

# Statistical modules



## Zonal statistics

r.statistics in 6.4 works with integer maps only (floating point ok in 6.5 and 7.0)

```
> g.region -p rast=elev_ned_30m@PERMANENT
> r.mapcalc "elev_ned_30m.int =
int(round(elev_ned_30m@PERMANENT))"
> g.region -p rast=landclass96@PERMANENT
> r.statistics base=landclass96@PERMANENT
cover=elev_ned_30m.int method=average
output=elevation_by_landclass96
```

# Statistical modules



## Raster map series

command: `r.series`

wxGUI: Raster -> Overlay rasters -> Raster series

# Statistical modules



## Raster map series

r.series

```
> g.region -p rast=precip_jan  
> r.series input=`g.mlist pattern=prec*  
separator=comma`  
output=precip_mean,precip_median,precip_stddev  
method=average,median,stddev
```

# Statistical modules



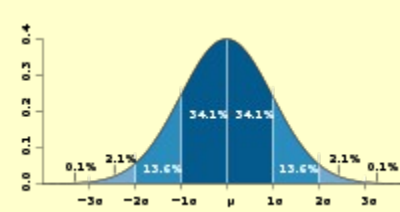
## Raster map series

r.series

With wxGUI:

1. File -> Manage maps and volumes -> List filtered (g.mlist)
2. Raster -> Overlay rasters -> Raster series (r.series)
3. Copy output of g.mlist to input of r.series
4. Define output map names and aggregate operation options

# Statistical modules



## Statistics combining raster and vector maps

command: `v.rast.stats`

wxGUI: Vector -> Update area attributes from raster

# Statistical modules



## Statistics combining raster and vector maps

`v.rast.stats`

First make a copy of vector `boundary_county@PERMANENT`  
set region to `elev_state_500m@PERMANENT`

```
> g.copy  
vect=boundary_county@PERMANENT,my_boundary_county  
  
> g.region -p rast=elev_state_500m@PERMANENT  
  
> v.rast.stats vector=my_boundary_county  
raster=elev_state_500m@PERMANENT colprefix=elev
```

# Statistical modules



## Statistics on one vector

command: `v.univar`, `v.db.univar` for attribute columns only

wxGUI: Vector  $\rightarrow$  Reports and statistics  $\rightarrow$  Univariate attribute statistics for points

# Statistical modules



## Statistics on one vector

`v.univar`

`compare`

```
> v.univar map=boundary_county@PERMANENT  
type=centroid column=PERIMETER
```

with

```
> v.univar map=boundary_county@PERMANENT  
type=boundary column=PERIMETER
```

# GRASS + R



## several possibilities

- Direct GRASS – R interface with R package spgrass6
- R package rgdal
- Calculate test statistics with GRASS,  
test for significance with R

# End of the seminar

Thank you for your interest and your attention!

Markus Metz and Markus Neteler

<http://grass.osgeo.org>





# License of this document

This work is licensed under a Creative Commons License.

<http://creativecommons.org/licenses/by-sa/2.5/deed.en>



“GIS seminar: An Introduction to GRASS”,  
© 2010 Markus Metz, Germany

## **License details: Attribution-ShareAlike 2.5**

You are free:

- to copy, distribute, display, and perform the work,
- to make derivative works,
- to make commercial use of the work,

under the following conditions:

Attribution. You must give the original author credit.

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

For any reuse or distribution, you must make clear to others the license terms of this work. Any of these conditions can be waived if you get permission from the copyright holder. Your fair use and other rights are in no way affected by the above.