I started my acquaintance with the Julia language by analyzing fictitious medical research. I chose the topic of healthcare because I would like to continue my thesis submitted at the bachelor's program, which examined the digitalization development of a nursing home, and the Julia language proves to be a good tool for this purpose.

Sample drawn from population admitted with skin and soft tissue infection due to diabetes mellitus

Classify infections into two groups by definiton:

```
1 - Minor infection
2 - Major infection
```

Descriptive statistics:

```
1. Count of number per group of infection
2. Count of number per group of gender
3. Mean of age
4. Summary of HbA1c and CRP
5. Age analysis and age distribution by type of infection
6. Age analysis and age distribution by type of gender
7. HbA1c analysis by type of infection and by gender
8. CRP analysis by type of infection and by gender
```

Importing packages

In [1]:
```
using Pkg
```

In [2]:
```
Pkg.add("IJulia")
Pkg.add("DataFrames")
Pkg.add("Gadfly")
Pkg.add("StatsBase")
Pkg.add("Distributions")
Pkg.add("CSV")
```

```
    Updating registry at `C:\Users\Lenovo ThinkPad T450\.julia\registries\General.toml`
   Resolving package versions...
  No Changes to `C:\Users\Lenovo ThinkPad T450\.julia\environments\v1.9\Project.toml`
  No Changes to `C:\Users\Lenovo ThinkPad T450\.julia\environments\v1.9\Manifest.toml`
   Resolving package versions...
  No Changes to `C:\Users\Lenovo ThinkPad T450\.julia\environments\v1.9\Project.toml`
  No Changes to `C:\Users\Lenovo ThinkPad T450\.julia\environments\v1.9\Manifest.toml`
   Resolving package versions...
  No Changes to `C:\Users\Lenovo ThinkPad T450\.julia\environments\v1.9\Project.toml`
  No Changes to `C:\Users\Lenovo ThinkPad T450\.julia\environments\v1.9\Manifest.toml`
   Resolving package versions...
  No Changes to `C:\Users\Lenovo ThinkPad T450\.julia\environments\v1.9\Project.toml`
  No Changes to `C:\Users\Lenovo ThinkPad T450\.julia\environments\v1.9\Manifest.toml`
   Resolving package versions...
  No Changes to `C:\Users\Lenovo ThinkPad T450\.julia\environments\v1.9\Project.toml`
  No Changes to `C:\Users\Lenovo ThinkPad T450\.julia\environments\v1.9\Manifest.toml`
   Resolving package versions...
  No Changes to `C:\Users\Lenovo ThinkPad T450\.julia\environments\v1.9\Project.toml`
  No Changes to `C:\Users\Lenovo ThinkPad T450\.julia\environments\v1.9\Manifest.toml`
```

In [3]:
```
using IJulia
using DataFrames
```

```
using Gadfly
using StatsBase
using Distributions
using CSV
```

Import dataset

In [4]:
```
df = DataFrame(CSV.File("Raw_data.csv"))
```

Out[4]: 120×6 DataFrame

95 rows omitted

Row	PatientID	Cat1	Cat2	Var1	Var2	Var3
	Int64	String1	String1	Float64	Float64	Float64
1	1	A	C	38.2568	5.93913	35.0579
2	2	A	C	17.8317	5.34754	21.131
3	8	A	B	16.0218	6.60709	60.9436
4	9	A	C	45.1158	6.00733	21.8797
5	16	A	C	20.448	8.54819	20.6623
6	18	A	B	28.3549	7.95642	33.1807
7	25	A	C	22.4497	6.34618	40.2365
8	28	A	B	48.4125	5.32583	28.8956
9	29	A	C	40.0075	11.4189	71.5911
10	33	A	C	20.7181	5.37768	27.4216
11	37	A	B	17.0396	5.34168	24.3501
12	38	A	B	42.6687	5.82284	52.361
13	41	A	B	19.954	5.13911	93.1999
⋮	⋮	⋮	⋮	⋮	⋮	⋮
109	76	B	R	17.0029	5.39477	31.1297
110	78	B	L	55.3879	4.15304	40.0846
111	80	B	R	20.2205	6.36442	45.435
112	83	B	L	16.4172	3.84167	89.6969
113	84	B	R	47.6224	5.40032	47.4541
114	86	B	L	73.0229	3.38349	55.1737
115	92	B	R	16.4106	4.30351	87.7357
116	95	B	R	16.2801	3.37252	52.6018
117	101	B	L	16.8883	3.19598	60.1883
118	113	B	R	32.3537	3.38677	30.0157
119	115	B	R	20.1379	3.42731	44.6893
120	119	B	L	17.6144	3.45116	40.6947

In [5]:
```
#Making sure there are no NA-values and looking at the data types
describe(df)
```

Out[5]: 6×7 DataFrame

Row	variable	mean	min	median	max	nmissing	eltype
	Symbol	Union...	Any	Union...	Any	Int64	DataType
1	PatientID	60.5	1	60.5	120	0	Int64
2	Cat1		A		B	0	String1
3	Cat2		B		X	0	String1
4	Var1	27.9679	15.2356	22.6801	84.2378	0	Float64
5	Var2	5.92121	3.01173	5.64241	15.5826	0	Float64
6	Var3	51.95	20.3153	44.3042	147.397	0	Float64

Changing coded values

In [6]:
```
#Changing the values of Cat1
# A was Minor infections
# B was Major infections
nrows, ncols = size(df)
for r in 1:nrows # Loop through all the rows
    infection_var = df[r, :Cat1] # variable creation
    if ismissing(infection_var)
        elseif infection_var == "A"
        df[!,:Cat1] = convert.(String31,df[!,:Cat1])
        df[r, :Cat1] = "Minor infection"
        elseif infection_var == "B"
        df[!,:Cat1] = convert.(String31,df[!,:Cat1])
        df[r, :Cat1] = "Major infection"
    else
    end
end
```

In [7]:
```
#Changing the values of Cat2
for r in 1:nrows
    gender_var = df[r, :Cat2]
    if ismissing(gender_var)
        elseif gender_var == "C" || gender_var == "X" || gender_var == "R" # OR
        df[!,:Cat2] = convert.(String15,df[!,:Cat2])
        df[r, :Cat2] = "Female"
        elseif gender_var == "L" || gender_var == "B" || gender_var == "F"
        df[!,:Cat2] = convert.(String15,df[!,:Cat2])
        df[r, :Cat2] = "Male"
    else
    end
end
```

In [8]:
```
#Changing the values of Var1
const fractional_digits = 0
for r in 1:nrows
    age_var = df[r, :Var1]
    if ismissing(age_var)
    else
```

```
        df[!,:Var1] = floor.(df[:,:Var1], digits=fractional_digits)
    end
end
```

In [9]:
```
# Renaming the columns
rename!(df, [:Cat1, :Cat2, :Var1, :Var2, :Var3] .=> [:Infection, :Gender, :Age, :HbA1c, :CRP])
first(df)
```

Out[9]: DataFrameRow (6 columns)

Row	PatientID	Infection	Gender	Age	HbA1c	CRP
	Int64	String31	String15	Float64	Float64	Float64
1	1	Minor infection	Female	38.0	5.93913	35.0579

Descriptive statistics

In [10]:
```
# 1. Count of number per group of infection
inf_groups = combine(groupby(df, :Infection), d -> DataFrame(Number = size(d,1)))
```

Out[10]: 2×2 DataFrame

Row	Infection	Number
	String31	Int64
1	Minor infection	60
2	Major infection	60

In [11]:
```
# 2. Count of number per group of gender
gender_groups = combine(groupby(df, :Gender), d -> DataFrame(Number = size(d,1)))
```

Out[11]: 2×2 DataFrame

Row	Gender	Number
	String15	Int64
1	Female	60
2	Male	60

In [12]:
```
# 3. Mean of age
floor.(Int, mean(df.Age))
```

Out[12]: 27

HbA1c value	Metabolic state
≤5.6%	normal
5.7-6.4%	prediabetes
≥6.5%	diabetes

In [25]:
```
# 4. Summary of HbA1c and CRP
describe(df.HbA1c)
```

```
Summary Stats:
Length:         120
Missing Count:  0
Mean:           5.921205
Minimum:        3.011733
1st Quartile:   4.065523
Median:         5.642406
3rd Quartile:   6.839651
Maximum:        15.582649
Type:           Float64
```

In general we can say that a C-reactive protein level below 10mg/l is considered within the normal range, while - it can even reach 1000mg/l - we are talking about a high CRP value

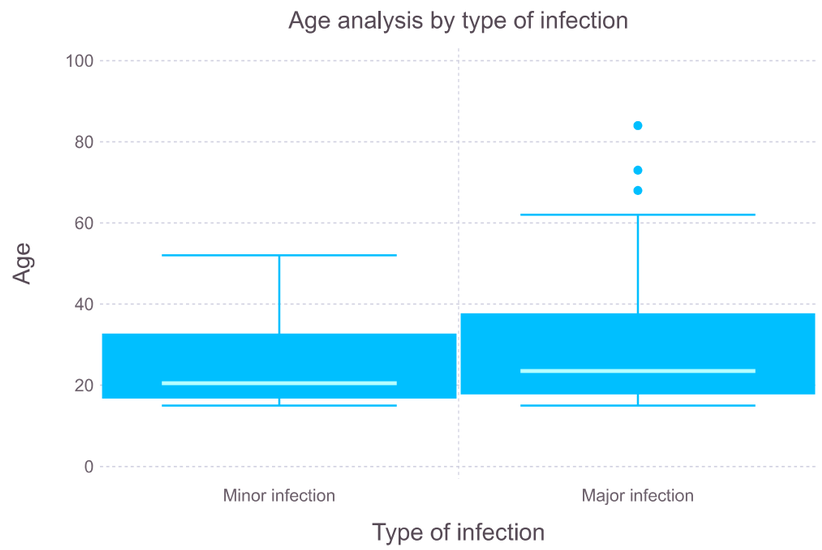
In [14]:
```
# 4. Summary of HbA1c and CRP
describe(df.CRP)
```

```
Summary Stats:
Length:         120
Missing Count:  0
Mean:           51.950031
Minimum:        20.315296
1st Quartile:   32.235514
Median:         44.304176
3rd Quartile:   64.858850
Maximum:        147.397402
Type:           Float64
```

1. Age analysis by type of infection:

In [26]:
```
#Using the Gadfly package
plot(df, x = "Infection", y = "Age", Geom.boxplot, Guide.title("Age analysis by type of infection"), Guide.xlabel("Type of infection"), Guide.ylabel("Age"))
```

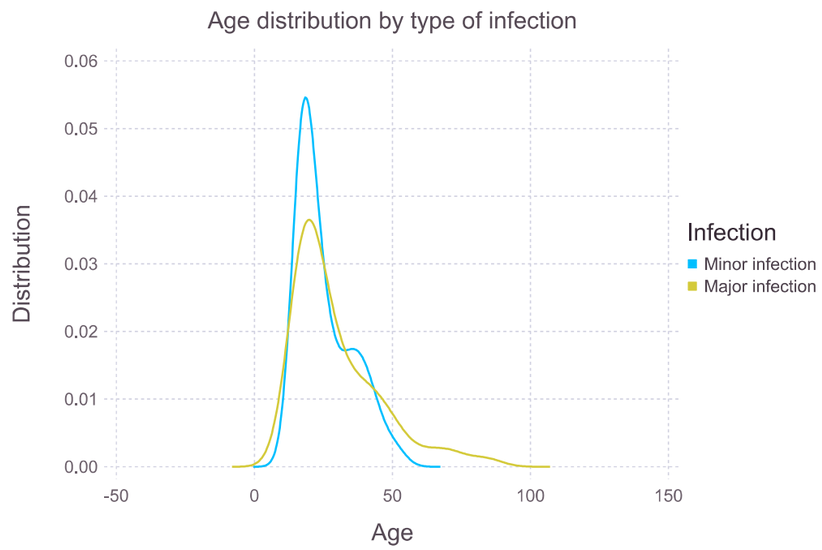
Out[26]:



Age distribution by type of infection:

In [27]:
```
plot(df, x = "Age", color = "Infection", Geom.density, Guide.title("Age distribution by type of infection"), Guide.xlabel("Age"), Guide.ylabel("Distribution"))
```

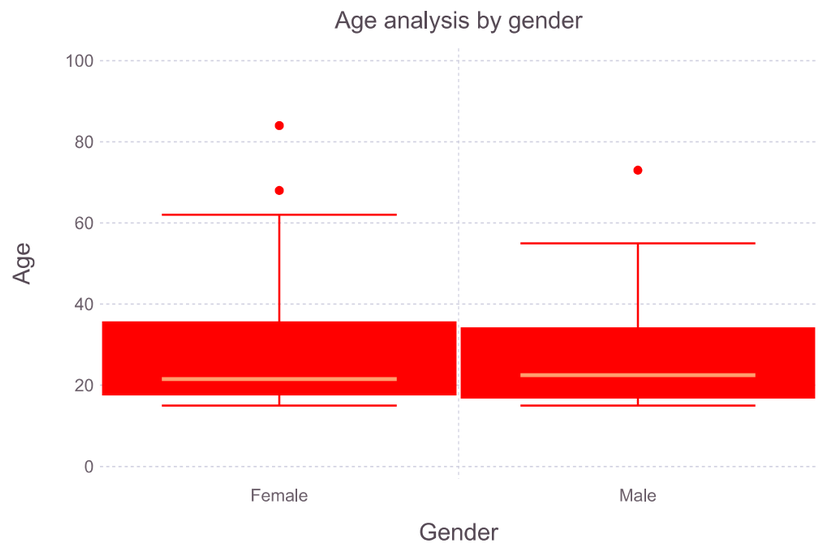
Out[27]:



1. Age analysis by gender:

In [17]:
```
plot(df, x = "Gender", y = "Age", Geom.boxplot, Guide.title("Age analysis by gender"), Guide.xlabel("Gender"), Guide.ylabel("Age"), Theme(default_color=colorant"red"))
```

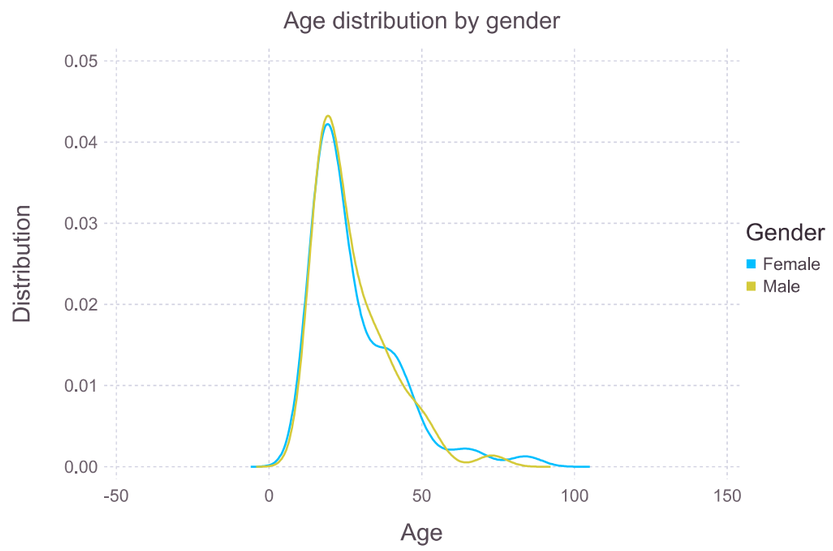
Out[17]:



Age distribution by gender:

In [19]:
```
plot(df, x = "Age", color = "Gender", Geom.density, Guide.title("Age distribution by gender"), Guide.xlabel("Age"), Guide.ylabel("Distribution"))
```

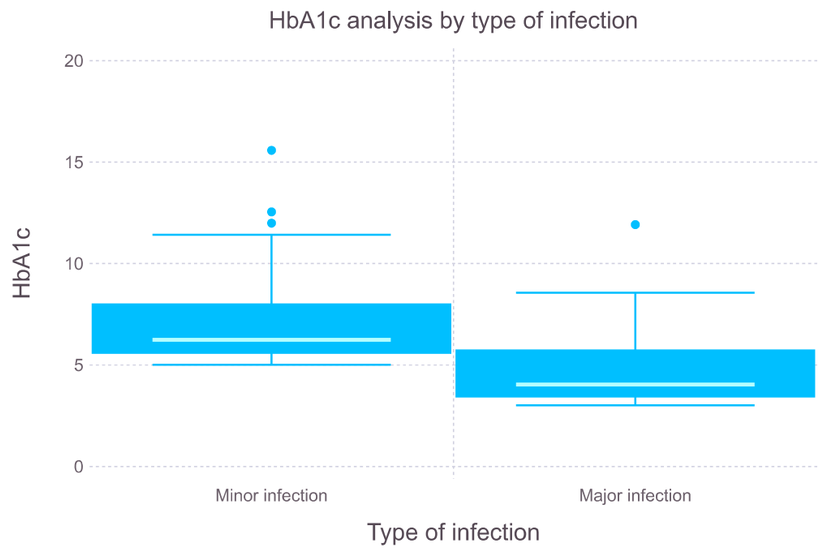
Out[19]:



1. HbA1c analysis by type of infection:

In [20]:
```
plot(df, x = "Infection", y = "HbA1c", Geom.boxplot, Guide.title("HbA1c analysis by type of infection"), Guide.xlabel("Type of infection"), Guide.ylabel("HbA1c"))
```

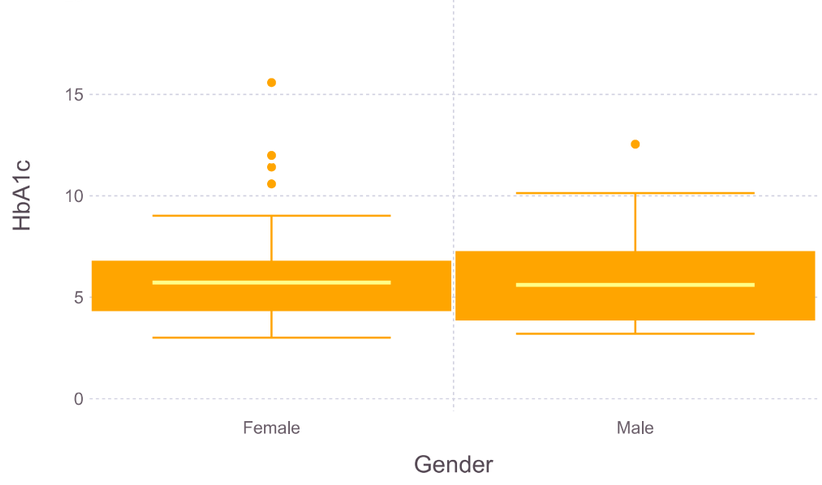
Out[20]:



HbA1c analysis by gender:

In [22]:
```
plot(df, x = "Gender", y = "HbA1c", Geom.boxplot, Guide.title("HbA1c analysis by gender"), Guide.xlabel("Gender"), Guide.ylabel("HbA1c"), Theme(default_color = colorant"orange"))
```

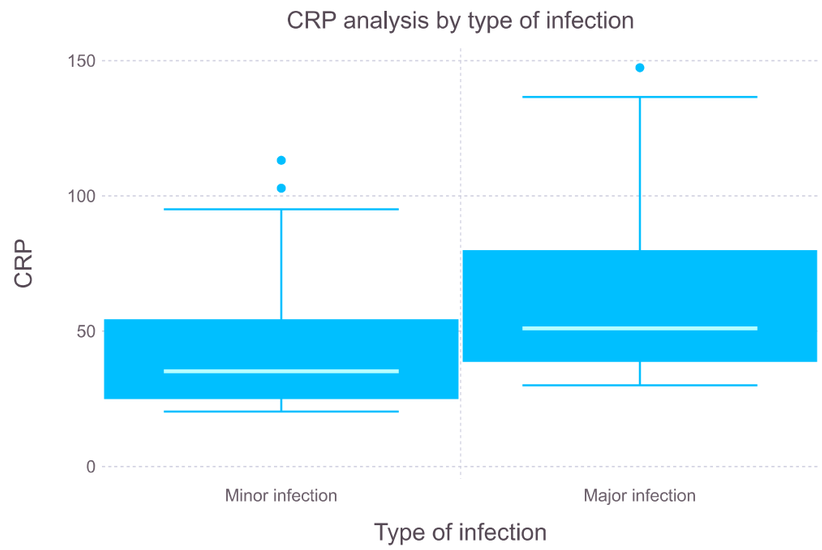
Out[22]:



1. CRP analysis by type of infection:

In [23]:
```
plot(df, x = "Infection", y = "CRP", Geom.boxplot, Guide.title("CRP analysis by type of infection"), Guide.xlabel("Type of infection"), Guide.ylabel("CRP"))
```

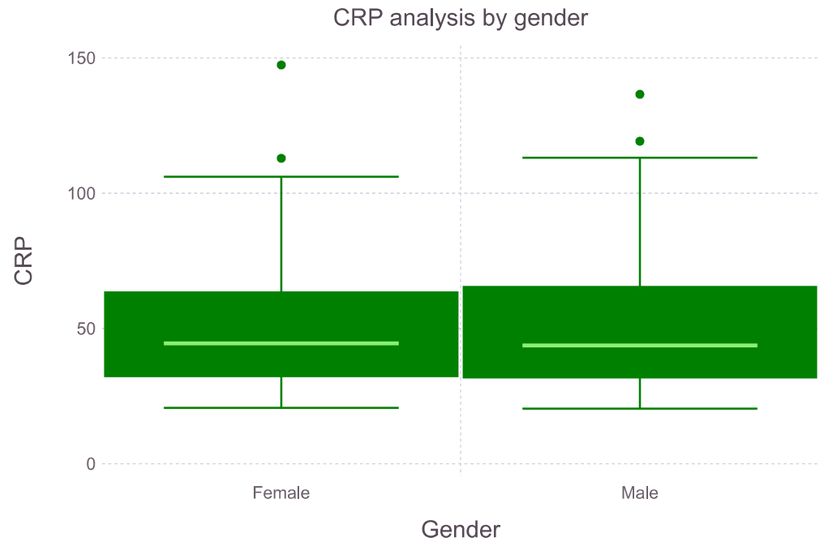
Out[23]:



CRP analysis by gender:

In [24]:
```
plot(df, x = "Gender", y = "CRP", Geom.boxplot, Guide.title("CRP analysis by gender"), Guide.xlabel("Gender"), Guide.ylabel("CRP"), Theme(default_color = colorant"green"))
```

Out[24]:



In [ ]: