Paper Code Documentation

This is the R Script corresponding to the Paper "Spatiotemporal patterns of cyclist collisions in Germany: variations in frequency, severity, and type of collisions in 2019". In the first section the functions for downloading and processing collision data from the collision dataset of the German Federal Statistical Office will be presented and explained step by step. The sections that follow represent the workflow for all statistics and graphs that are included in the paper.

Note: The library collisionsDE is a self written library by the leading author and a work in progress.

#Libraries

```
library("devtools")# for downloading GitHub packages
```

```
## Loading required package: usethis
```

```
#install_github("lutzhutz/collisionsDE")
library(collisionsDE)
#install.packages("colorspace")
library(colorspace)
#install.packages("dplyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#install.packages("sf")
library(sf)
```

```
## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1
```

```
#install.packages("ggplot2")
library(ggplot2)
#install.packages("lemon")
library(lemon)
```

```
##
## Attaching package: 'lemon'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     CoordCartesian, element_render
```

```
#install.packages("tidyr")
library(tidyr)
#install.packages("stringr")
```

```r
library(stringr)
#install.packages("ggpubr")
library(ggpubr)
#install.packages("grid")
library(grid)
#install.packages("lwgeom")
library(lwgeom)
```

```
## Linking to liblwgeom 3.0.0beta1 r16016, GEOS 3.8.0, PROJ 6.3.1
```

```r
#install.packages("readxl")
library(readxl)
```

Legend for data name extensions (chronologically):

- y.## = year (e.g. 2018 -> y.19)
- .df = data frame of all collisions
- .b = all collisions with at least one cyclist involved
- .f = frequency
- .k = killed
- .si = seriously injured
- .sli = slightly injured
- .sev = severity
- .ct = collision type
- .ck = collision - killed
- .c = collisions
- .m = month
- .h = hours (daytime)
- .w = weekdays
- .osm = highway data from Open Street Map
- .s = subset
- .m = maxspeed categories
- .mot = motorized
- .g = grouped
- .wi = with information (roads with information on maximum speed limit)
- .ni = no information (roads with no information on maximum speed limit)
- .mfc = merged feature classes (merged datasets of .wi and .ni)

#Pre-processing

```r
#disable e format
options(scipen = 999)

#import all reported collisions with personal injury from 2018 (function from collisionsDE)
#y.19<-import_2018()
y.19<-import_2019()

#subset all accident with at least one cyclist involved
y.19.b<- y.19[y.19$IstRad == 1,]

#add regional information to the collision events (function from collisionsDE)
y.19<-add_regions(y.19)
```

```
## Warning in add_regions(y.19): 235collisions could not be assigned a regional
## spatial type
```

```
y.19.b<-add_regions(y.19.b)
```

```
## Warning in add_regions(y.19.b): 18collisions could not be assigned a regional
## spatial type
```

```
#write_sf(y.19.b,"C:/Users/LH/Desktop/Uni/B.Sc/OSM_Highway_19/Unfall_19_proj.shp","Unfall_19_b")

#transform from spatial feature object to data frame
y.19.df.b <- y.19.b %>% st_drop_geometry()

y.19.df <- y.19 %>% st_drop_geometry()
#subset all accident with at least one cyclist involved
#y.19.df.b <- y.19.df[y.19.df$IstRad == 1,]

#add regional information to the collision events (function from collisionsDE)
#y.19.df.b<-add_regions(y.19.df.b)

y.19.df.b<-y.19.df.b[!is.na(y.19.df.b$regio7bez),]

y.19.df<-y.19.df[!is.na(y.19.df$regio7bez),]

#data frame overview
head(y.19.df.b)
```

```
##       gem OBJECTID ULAND UREGBEZ UKREIS UGEMEINDE UJAHR UMONAT USTUNDE
## 1 1001000   155354    01       0     01       000  2019     05      11
## 2 1001000     2819    01       0     01       000  2019     07      07
## 3 1001000   155986    01       0     01       000  2019     08      13
## 4 1001000     4451    01       0     01       000  2019     08      20
## 5 1001000   157224    01       0     01       000  2019     11      09
## 6 1001000     1739    01       0     01       000  2019     04      16
##   UWOCHENTAG UKATEGORIE UART UTYP1 ULICHTVERH IstRad IstPKW IstFuss IstKrad
## 1          4          3    5     2          0      1      1       0       0
## 2          3          3    1     5          0      1      1       0       0
## 3          7          3    0     7          0      1      0       0       0
## 4          4          3    0     7          1      1      0       0       0
## 5          7          3    0     7          0      1      0       0       0
## 6          2          3    4     6          0      1      1       0       0
##   IstGkfz IstSonstig  LINREFX LINREFY XGCSWGS84 YGCSWGS84 STRZUSTAND     gemrs
## 1       0          0 528096.6 6071086  9.436907  54.78618          0 1.001e+10
## 2       0          0 528549.3 6070974  9.443935  54.78515          0 1.001e+10
## 3       0          0 527481.3 6072846  9.427505  54.80204          1 1.001e+10
## 4       0          0 529576.0 6071760  9.459980  54.79215          1 1.001e+10
## 5       0          0 524594.7 6067805  9.382175  54.75688          2 1.001e+10
## 6       0          0 529557.8 6073552  9.459879  54.80826          0 1.001e+10
##            gemname gembev gemfl     vbg    vbgrs          vbgname land
## 1 Flensburg, Stadt  89504 56.73 1001000 10010000 Flensburg, Stadt    1
## 2 Flensburg, Stadt  89504 56.73 1001000 10010000 Flensburg, Stadt    1
## 3 Flensburg, Stadt  89504 56.73 1001000 10010000 Flensburg, Stadt    1
## 4 Flensburg, Stadt  89504 56.73 1001000 10010000 Flensburg, Stadt    1
## 5 Flensburg, Stadt  89504 56.73 1001000 10010000 Flensburg, Stadt    1
## 6 Flensburg, Stadt  89504 56.73 1001000 10010000 Flensburg, Stadt    1
##   RegioStaR2 RegioStaR4 RegioStaR17 RegioStaR7 RegioStaR5 RegioStaRGem7
## 1          2         22         221         75         54            74
```

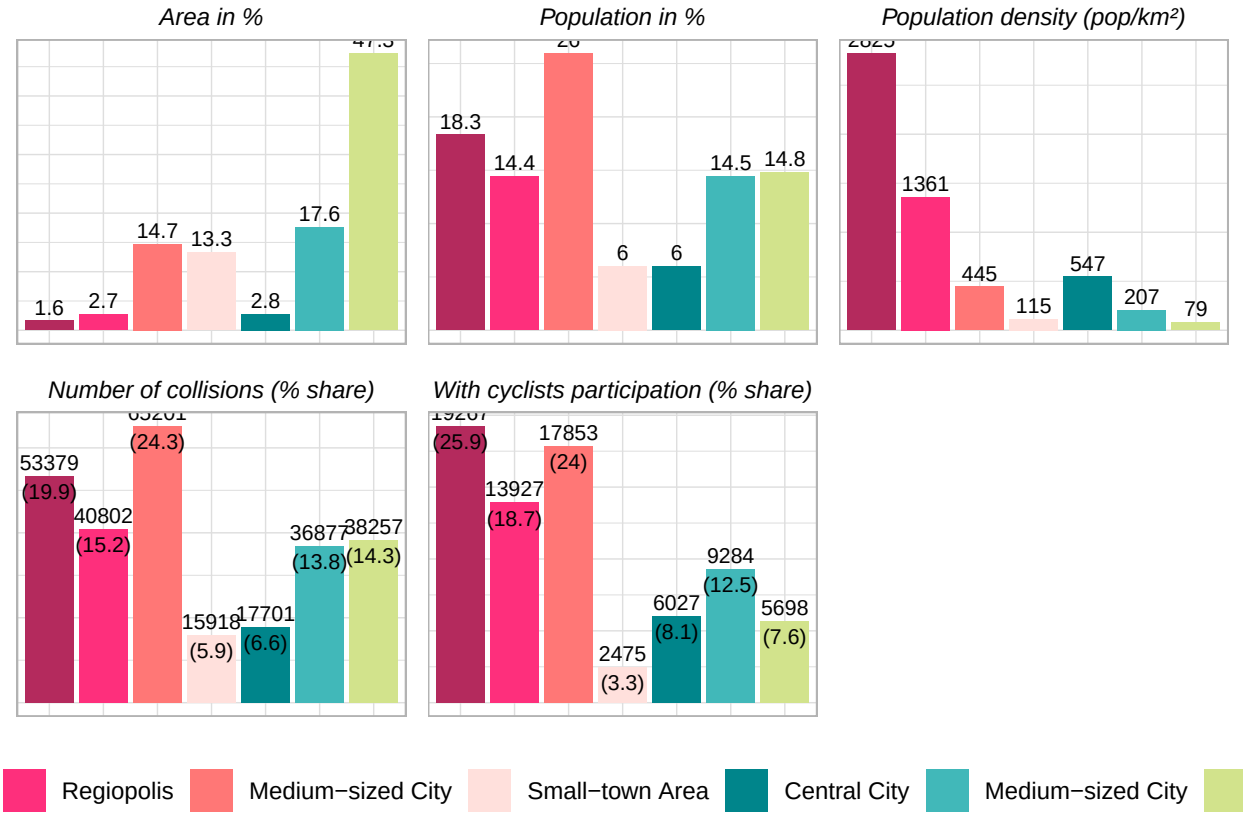
```
## 2          2         22         221         75         54            74
## 3          2         22         221         75         54            74
## 4          2         22         221         75         54            74
## 5          2         22         221         75         54            74
## 6          2         22         221         75         54            74
##   RegioStaRGem5 Stadtregion nameStadtregion regio7bez
## 1            53          NA            <NA>       R_C
## 2            53          NA            <NA>       R_C
## 3            53          NA            <NA>       R_C
## 4            53          NA            <NA>       R_C
## 5            53          NA            <NA>       R_C
## 6            53          NA            <NA>       R_C
```

```
#suppress note message from dplyr
options(dplyr.summarise.inform=F)

#y.19$ID <- seq.int(nrow(y.19))

#write_sf(y.19,"C:/Users/LH/Desktop/Uni/B.Sc/OSM_Highway_19/Unfall_19_proj.shp","Unfall_19_proj")
```

#General overview

```
#data transform
y.19.o <-
  y.19.df %>%
  mutate(countb = if_else(y.19.df$IstRad == 1, 1, 0)) %>%
  group_by(gemname, regio7bez, gembev, gemfl) %>%
  summarise(count = sum(n()), countb = sum(countb)) %>%
  group_by(regio7bez) %>%
  summarise(
    gsm_fl = sum(gemfl),
    gsm_bev = sum(gembev),
    count = sum(count),
    countb = sum(countb),
    popdens = round(gsm_bev / gsm_fl,0),
  ) %>%
  mutate(
    gsm_fl = gsm_fl / sum(gsm_fl) * 100,
    gsm_bev = gsm_bev / sum(gsm_bev) * 100
    #shareb = countb / sum(countb) * 100
  ) %>%
  pivot_longer(c(gsm_fl, gsm_bev, count, countb, popdens, gsm_fl, gsm_bev))

manlabel<-c("","","(19.9)","(25.9)","","","","(15.2)","(18.7)","","","","(24.3)","(24)","","","","(5.9)

#plot export in 1920x (maintain ratio)
ggplot(transform(y.19.o, name = factor(
  name,
  levels = c("gsm_fl", "gsm_bev", "popdens", "count", "countb"),
  labels = c(
    "Area in %",
    "Population in %",
    "Population density (pop/km²)",
    "Number of collisions (% share)",
    "With cyclists participation (% share)"
```

```
  )
)),
aes(regio7bez, value, fill = regio7bez)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = round(value, 1)), vjust = -0.4, size = 2.7) +
  geom_text(aes(label = manlabel), vjust = 1.5, size = 2.7)+
  facet_wrap(~ name, scales = "free") +
  theme_light()+
  theme(
    axis.title = element_blank(),
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank(),
    legend.position = "bottom",
    legend.title = element_blank(),
    legend.text = element_text(size = 9),
    strip.text = element_text(color = 'black',size = 9, face = "italic"),
    strip.background = element_blank()
  ) +
  scale_fill_manual(
    "legend",
    values = c(
      "#B42A5D",
      "#FE2F7C",
      "#FF7776",
      "#FFE0DC",
      "#00858B",
      "#41B8B9",
      "#D2E38C"
    ),
    labels = c(
      "Metropolis",
      "Regiopolis",
      "Medium-sized City",
      "Small-town Area",
      "Central City",
      "Medium-sized City",
      "Small-town Area"
    )
  ) +
  guides(fill = guide_legend(nrow = 1))
```




```
#save in output workspace
#ggsave("overview.png",dpi = 300,width = 8.83,height = 7.15)
```

#Collision severity (Figure 4)

```
y.19.df.k <-
  y.19.df.b[y.19.df.b$UKATEGORIE == 1, ] #subset all collisions with at least one person killed

#transform data
y.19.df.k <- y.19.df.k %>%
  group_by(regio7bez) %>%
  summarise(count = n()) %>%
  mutate(share = count / sum(count) * 100,
         cat = "killed") #calculate the shares of all fatal crashes by each region

y.19.df.si <-
  y.19.df.b[y.19.df.b$UKATEGORIE == 2, ] #subset all collisions with at least one person seriously inju

#transform data
y.19.df.si <- y.19.df.si %>%
  group_by(regio7bez) %>%
  summarise(count = n()) %>%
  mutate(share = count / sum(count) * 100,
         cat = "seriously_injured") #calculate the shares of all crashes with at least one person serio

y.19.df.sli <-
  y.19.df.b[y.19.df.b$UKATEGORIE == 3, ] #subset all collisions with at least one person slightly injur
```

```r
#transform data
y.19.df.sli <- y.19.df.sli %>%
  group_by(regio7bez) %>%
  summarise(count = n()) %>%
  mutate(share = count / sum(count) * 100,
         cat = "injury") #calculate the shares of all crashes with at least one person slightly injured


y.19.df.sev <-
  rbind(y.19.df.k, y.19.df.si, y.19.df.sli) #combine the three created datasets

#check total counts
sum(y.19.df.k$count)
```

```
## [1] 382
```

```r
sum(y.19.df.si$count)
```

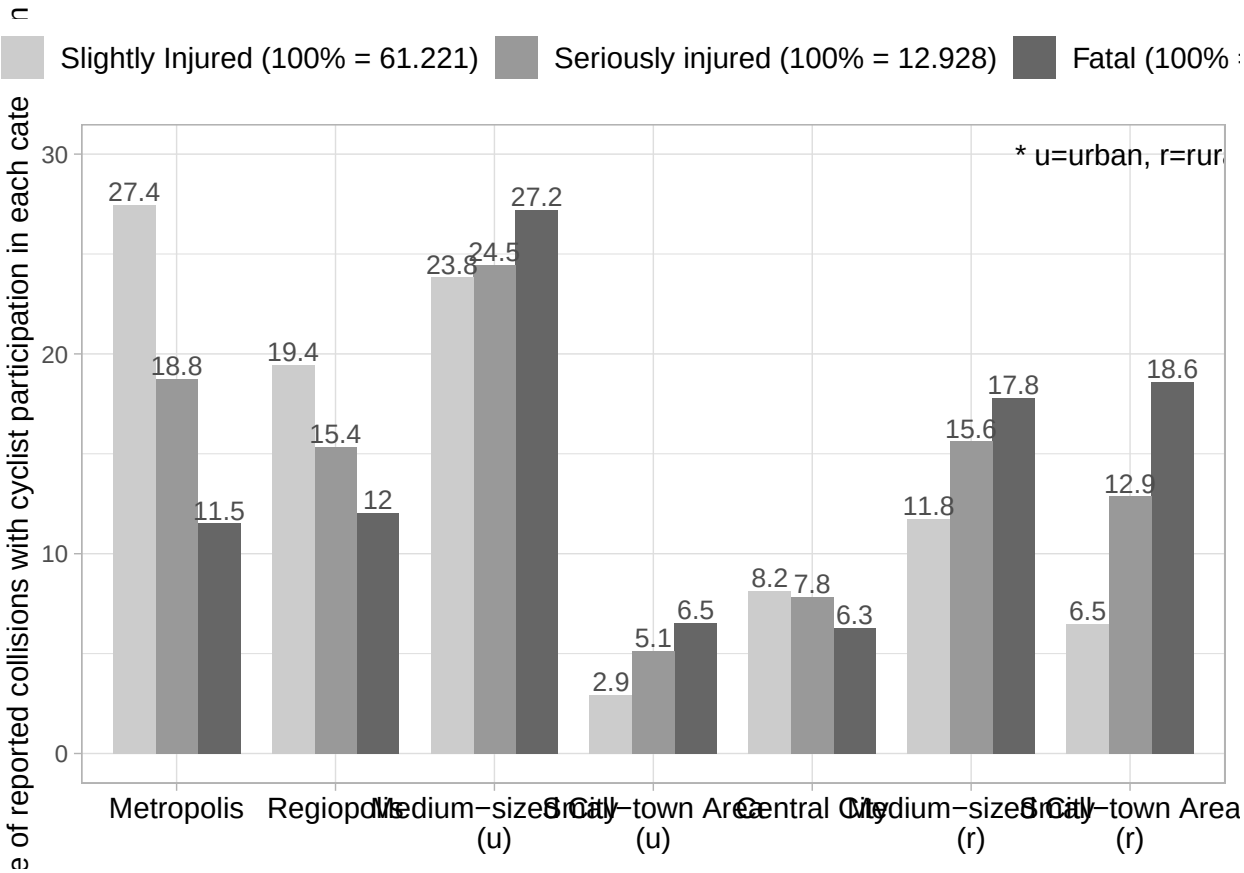
```
## [1] 12928
```

```r
sum(y.19.df.sli$count)
```

```
## [1] 61221
```

```r
labels = c(
        "Metropolis",
        "Regiopolis",
        "Medium-sized City\n(u)",
        "Small-town Area\n(u)",
        "Central City",
        "Medium-sized City\n(r)",
        "Small-town Area\n(r)"
      )



#visualization (Figure 4 in Paper)
ggplot(
  transform(
    y.19.df.sev,
    cat = factor(
      cat,
      levels =
        c("injury",
          "seriously_injured",
          "killed"),
      label = c(
        "Slightly Injured (100% = 61.221)",
        "Seriously injured (100% = 12.928)",
        "Fatal (100% = 382)"
      )
    ),
    regio7bez = factor(
      regio7bez,
      levels =
```

```
        c(
          "U_Metro",
          "U_Regiop",
          "U_Medium",
          "U_Small",
          "R_C",
          "R_Med_C",
          "R_Small_C"
        ),
      label = labels
    )
  ),
  aes(regio7bez, share, fill = cat)
) +
  geom_bar(stat = "identity",
           position = "dodge",
           width = 0.8) +
  geom_text(
    aes(label = round(share, 1)),
    position = position_dodge(width = 0.8),
    vjust = -0.2,
    color = "gray30",
    size = 3.5
  ) +
  ylab("Share of reported collisions with cyclist participation in each category in %") +
  theme_light() +
  scale_fill_manual("",
                    values = c("grey80",
                               "grey60",
                               "grey40")) +
  theme(
    axis.title.x = element_blank(),
    axis.text.x = element_text(colour = "black", size = 10.5),
    legend.position = "top",
    legend.justification = "left",
    legend.text = element_text(size = 10.5)
  ) +
  annotate("text",
           label = "* u=urban, r=rural",
           x = 7,
           y = 30)
```




```
#save in output workspace
#ggsave("severity.png", dpi = 300, width = 9.1, height = 7.15)
```

#Collision type (Figure 5)

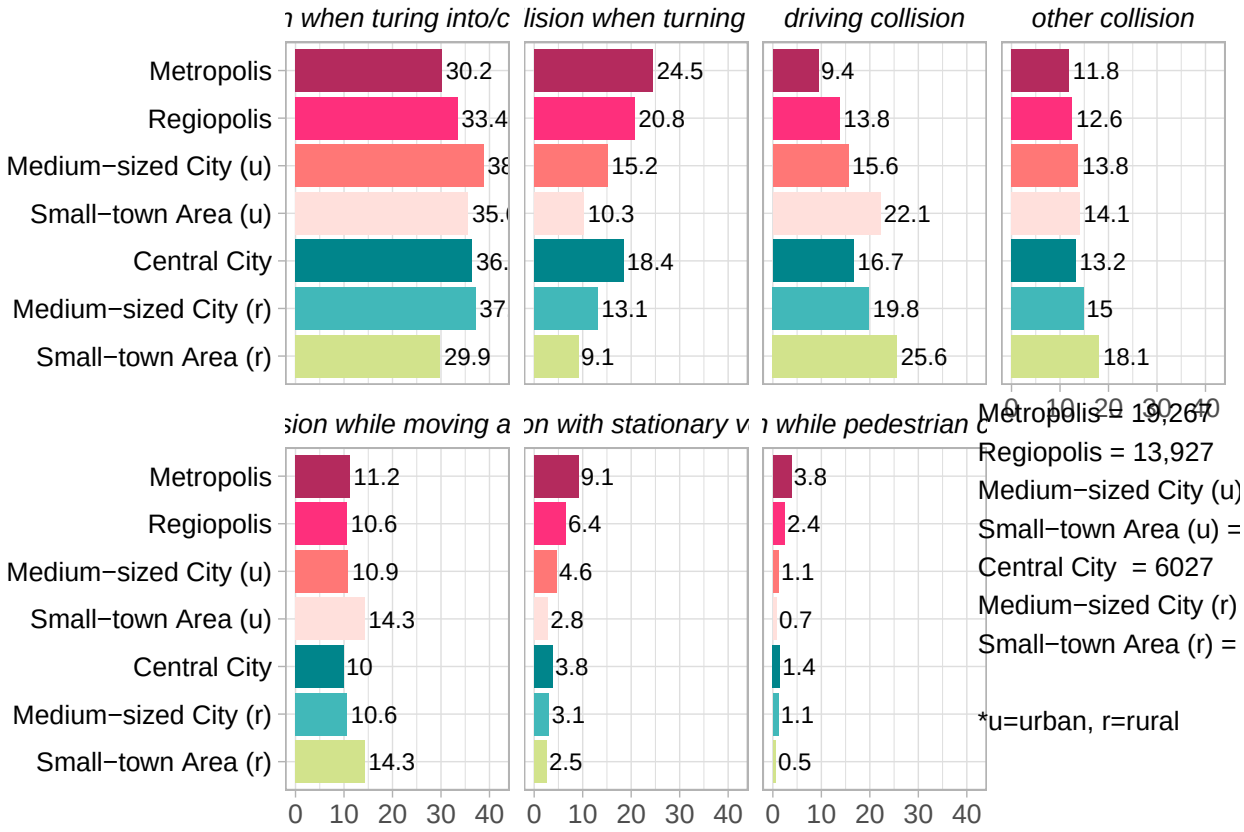
```
#calculate share of each collision type by region
y.19.df.b.ct <- y.19.df.b %>%
  group_by(regio7bez, UTYP1) %>%
  summarise(count = n()) %>%
  mutate(share = count / sum(count) * 100)

#visualize (Figure 5 in paper)
type<-ggplot(
    transform(
    y.19.df.b.ct,
    UTYP1 = factor(
      UTYP1,
      levels = c(
        "3",
        "2",
        "1",
        "7",
        "6",
        "5",
        "4"
      ),
      label = c(
```

```
        "collision when turing into/crossing",
        "collision when turning off",
        "driving collision",
        "other collision",
        "collision while moving along",
        "collision with stationary vehicle",
        "collision while pedestrian crossing"
      )
    ),
    regio7bez = factor(
      regio7bez,
      levels =
        c(
          "R_Small_C",
          "R_Med_C",
          "R_C",
          "U_Small",
          "U_Medium",
          "U_Regiop",
          "U_Metro"
        ),
      label = c(
        "Small-town Area (r)",
        "Medium-sized City (r)",
        "Central City",
        "Small-town Area (u)",
        "Medium-sized City (u)",
        "Regiopolis",
        "Metropolis"
      )
    )
  ),
  aes(x = share, y = regio7bez)
) +
  facet_wrap( ~ UTYP1, ncol = 4)  +
  geom_bar(stat = "identity", fill = rep(c("#B42A5D",
    "#FE2F7C",
    "#FF7776",
    "#FFE0DC",
    "#00858B",
    "#41B8B9",
    "#D2E38C"
  ),7)) +
  geom_text(aes(label = round(share,1)), hjust = -0.08, size = 3) +
  theme_light() +
  scale_x_continuous(limits = c(0, 42)) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(color = 'black',size = 10),
    strip.text = element_text(color = 'black',size = 10, face = "italic"),
    strip.background = element_blank()
```

```
  )

#check total counts
y.19.df.b.ct %>%
  group_by(regio7bez) %>%
  summarise(count = sum(count))
```

```
## # A tibble: 7 x 2
##   regio7bez count
##   <fct>     <int>
## 1 U_Metro   19267
## 2 U_Regiop  13927
## 3 U_Medium  17853
## 4 U_Small    2475
## 5 R_C        6027
## 6 R_Med_C    9284
## 7 R_Small_C  5698
```

```
#add counts to ggplot graph
print(type)
grid.text(
  paste0(
    "Metropolis = 19,267",
    "\n",
    "Regiopolis = 13,927",
    "\n",
    "Medium-sized City (u) = 17,853",
    "\n",
    "Small-town Area (u) = 2,475",
    "\n",
    "Central City  = 6027",
    "\n",
    "Medium-sized City (r) = 9,284",
    "\n",
    "Small-town Area (r) = 5,698",
    "\n",
    "\n",
    "*u=urban, r=rural"
  ),
  x = 0.79,
  y = 0.32,
  just = "left",
  gp = gpar(fontsize = 10)
)
```




```
#Export/Save as image -> 1122x590
```

#Collisions between cyclists and other parties with fatal outcome (Figure 6)

```
#subset all collisions with at least one person killed
y.19.df.b.ck <- y.19.df.b[y.19.df.b$UKATEGORIE == 1,]

#add collision type with collision_types() (function from collisionsDE)
y.19.df.b.ck <- collision_types(y.19.df.b.ck)

#group by region and collision type, compute shares
y.19.df.b.ck <- y.19.df.b.ck %>%
  group_by(regio7bez, coll_typ) %>%
  summarise(count = n()) %>%
  mutate(share = count / sum(count) * 100)

#visualize (Figure 6 in the paper)
b<-ggplot(
  transform(
    y.19.df.b.ck,
    coll_typ = factor(
      coll_typ,
      levels = c(
        "bicyclecar",
        "bicycle",
        "bicyclefoot",
        "bicycleother",
```

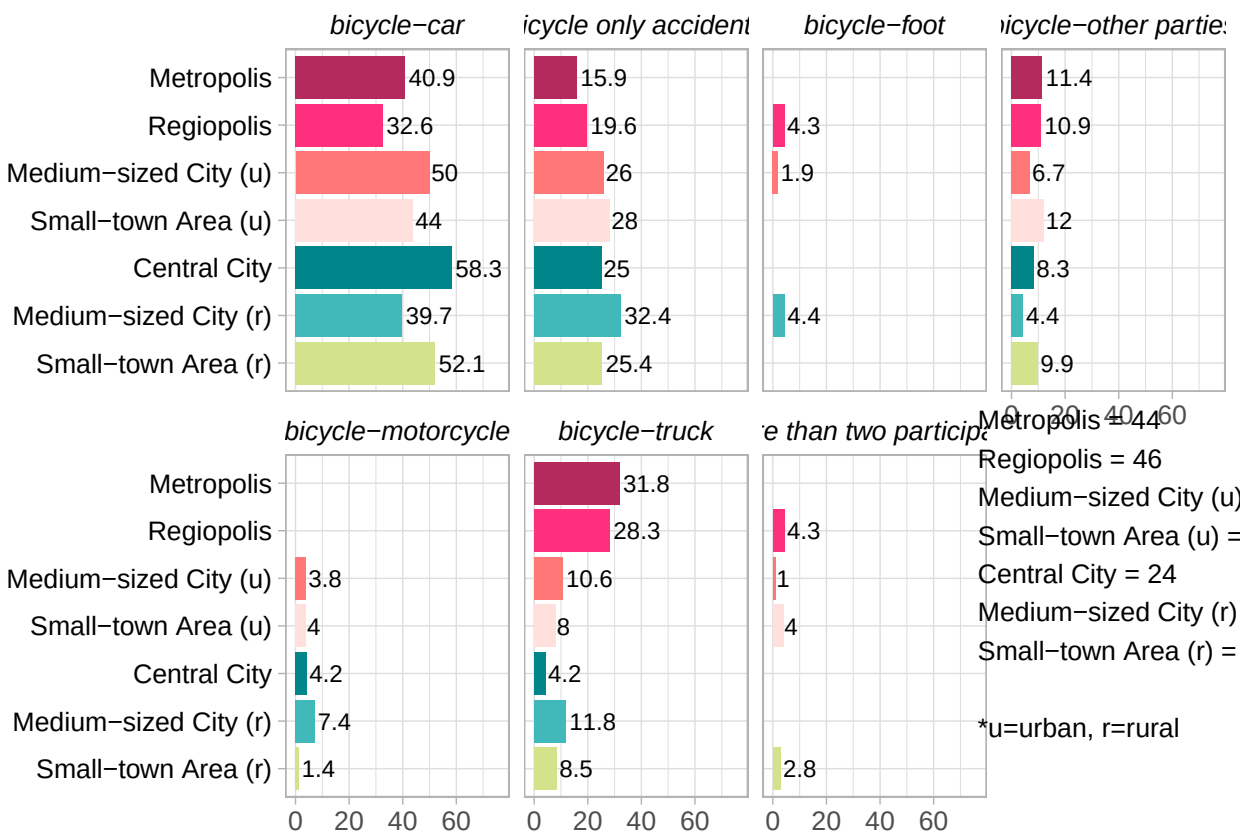
```
        "bicyclemcycle",
        "bicycletruck",
        "three"
      ),
      labels = c(
        "bicycle-car",
        "bicycle only accidents",
        "bicycle-foot",
        "bicycle-other parties",
        "bicycle-motorcycle",
        "bicycle-truck",
        "more than two participants"
      )
    ),
    regio7bez = factor(
      regio7bez,
      levels =
        c(
          "R_Small_C",
          "R_Med_C",
          "R_C",
          "U_Small",
          "U_Medium",
          "U_Regiop",
          "U_Metro"
        ),
      label = c(
        "Small-town Area (r)",
        "Medium-sized City (r)",
        "Central City",
        "Small-town Area (u)",
        "Medium-sized City (u)",
        "Regiopolis",
        "Metropolis"
      )
    )
  ),
  aes(x = share, y = regio7bez)
) +
  facet_wrap( ~ coll_typ, ncol = 4)  +
  geom_bar(
    stat = "identity",
    fill = c(
      "#B42A5D",#1
      "#FE2F7C",#2
      "#FF7776",#3
      "#FFE0DC",#4
      "#00858B",#5
      "#41B8B9",#6
      "#D2E38C",#7
      "#B42A5D",#1
      "#FE2F7C",#2
      "#FF7776",#3
```

```
      "#FFE0DC",#4
      "#00858B",#5
      "#41B8B9",#6
      "#D2E38C",#7
      "#FE2F7C",#2
      "#FF7776",#3
      "#41B8B9",#6
      "#B42A5D",#1
      "#FE2F7C",#2
      "#FF7776",#3
      "#FFE0DC",#4
      "#00858B",#5
      "#41B8B9",#6
      "#D2E38C",#7
      "#FF7776",#3
      "#FFE0DC",#4
      "#00858B",#5
      "#41B8B9",#6
      "#D2E38C",#7
      "#B42A5D",#1
      "#FE2F7C",#2
      "#FF7776",#3
      "#FFE0DC",#4
      "#00858B",#5
      "#41B8B9",#6
      "#D2E38C",#7
      "#FE2F7C",#2
      "#FF7776",#3
      "#FFE0DC",#4
      "#D2E38C"#7
    )
  ) +
  geom_text(aes(label = round(share, 1)), hjust = -0.08, size = 3) +
  theme_light() +
  scale_x_continuous(limits = c(0, 76)) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(color = 'black', size = 10),
    strip.text = element_text(color = 'black', size = 10, face = "italic"),
    strip.background = element_blank()
  )

#check total counts
y.19.df.b.ck %>%
  group_by(regio7bez) %>%
  summarise(count = sum(count))
```

```
## # A tibble: 7 x 2
##   regio7bez count
##   <fct>     <int>
## 1 U_Metro      44
## 2 U_Regiop     46
```

```
## 3 U_Medium    104
## 4 U_Small      25
## 5 R_C          24
## 6 R_Med_C      68
## 7 R_Small_C    71
```

```
#add counts to ggplot graph
print(b)
grid.text(
  paste0(
    "Metropolis = 44",
    "\n",
    "Regiopolis = 46",
    "\n",
    "Medium-sized City (u) = 104",
    "\n",
    "Small-town Area (u) = 25",
    "\n",
    "Central City = 24",
    "\n",
    "Medium-sized City (r) = 68",
    "\n",
    "Small-town Area (r) = 71",
    "\n",
    "\n",
    "*u=urban, r=rural"
  ),
  x = 0.79,
  y = 0.32,
  just = "left",
  gp = gpar(fontsize = 10)
)
```




```
#Export/Save as image -> 1122x590
```

#Collisions between cyclists and other parties, all collisions (Figure 7)

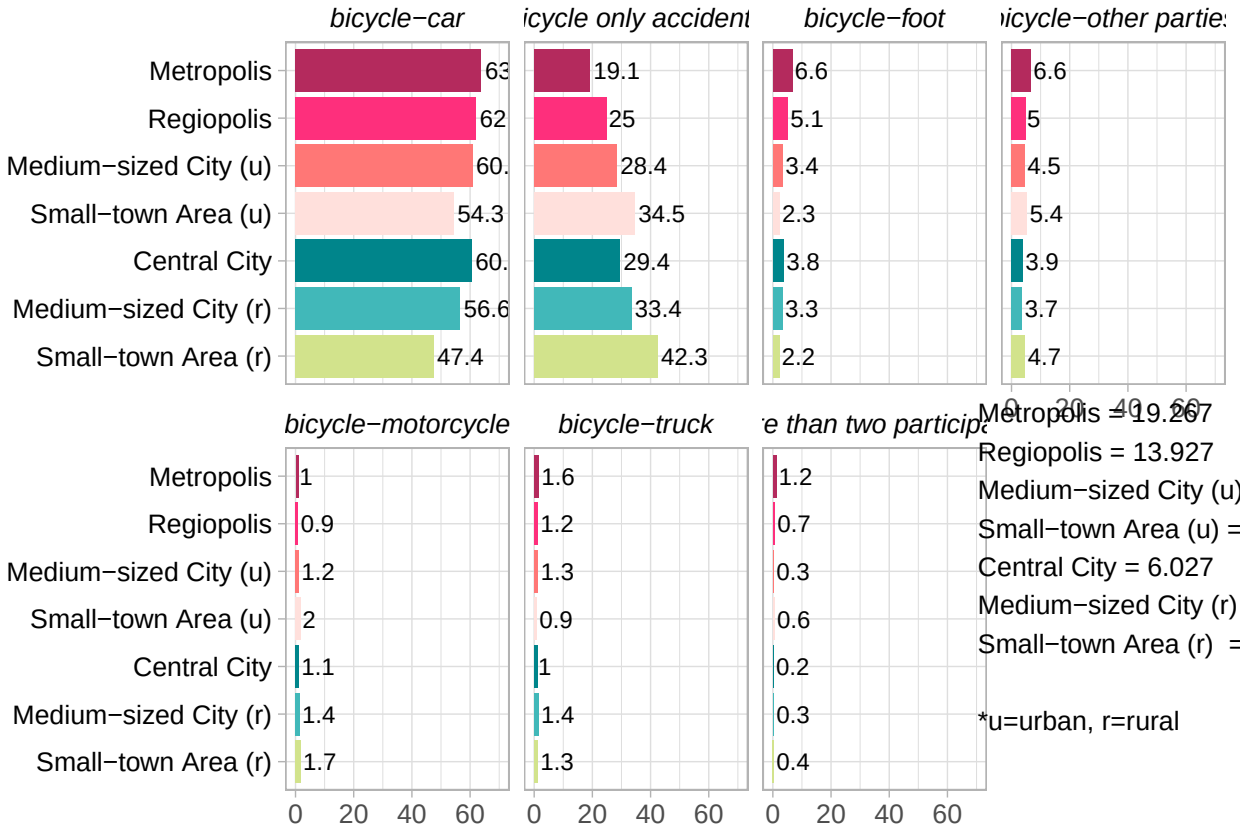
```
#add collision type with collision_types() function by collisionsDE
y.19.df.b.c <- collision_types(y.19.df.b)

#group by region and collision type, compute shares
y.19.df.b.c <- y.19.df.b.c %>%
  group_by(regio7bez, coll_typ) %>%
  summarise(count = n()) %>%
  mutate(share = count / sum(count) * 100)

#visualize (Figure 7 in the paper)
a <- ggplot(
  transform(
    y.19.df.b.c,
    coll_typ = factor(
      coll_typ,
      levels = c(
        "bicyclecar",
        "bicycle",
        "bicyclefoot",
        "bicycleother",
        "bicyclemcycle",
        "bicycletruck",
        "three"
```

```
      ),
      labels = c(
        "bicycle-car",
        "bicycle only accidents",
        "bicycle-foot",
        "bicycle-other parties",
        "bicycle-motorcycle",
        "bicycle-truck",
        "more than two participants"
      )
    ),
    regio7bez = factor(
      regio7bez,
      levels =
        c(
          "R_Small_C",
          "R_Med_C",
          "R_C",
          "U_Small",
          "U_Medium",
          "U_Regiop",
          "U_Metro"
        ),
      label = c(
        "Small-town Area (r)",
        "Medium-sized City (r)",
        "Central City",
        "Small-town Area (u)",
        "Medium-sized City (u)",
        "Regiopolis",
        "Metropolis"
      )
    )
  ),
  aes(x = share, y = regio7bez)
) +
  facet_wrap(~ coll_typ, ncol = 4)  +
  geom_bar(stat = "identity", fill = rep(
    c(
      "#B42A5D",
      "#FE2F7C",
      "#FF7776",
      "#FFE0DC",
      "#00858B",
      "#41B8B9",
      "#D2E38C"
    ),
    7
  )) +
  geom_text(aes(label = round(share, 1)), hjust = -0.08, size = 3) +
  theme_light() +
  scale_x_continuous(limits = c(0, 70)) +
  theme(
```

```
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(color = 'black', size = 10),
    strip.text = element_text(color = 'black', size = 10, face = "italic"),
    strip.background = element_blank()
  )

#check total counts
y.19.df.b.c %>%
  group_by(regio7bez) %>%
  summarise(count = sum(count))
```

```
## # A tibble: 7 x 2
##   regio7bez count
##   <fct>     <int>
## 1 U_Metro   19267
## 2 U_Regiop  13927
## 3 U_Medium  17853
## 4 U_Small    2475
## 5 R_C        6027
## 6 R_Med_C    9284
## 7 R_Small_C  5698
```

```
#add counts to ggplot graph
print(a)
grid.text(
  paste0(
    "Metropolis = 19.267",
    "\n",
    "Regiopolis = 13.927",
    "\n",
    "Medium-sized City (u) = 17.853",
    "\n",
    "Small-town Area (u) = 2.475",
    "\n",
    "Central City = 6.027",
    "\n",
    "Medium-sized City (r) = 9.284",
    "\n",
    "Small-town Area (r)  = 5.698",
    "\n",
    "\n",
    "*u=urban, r=rural"
  ),
  x = 0.79,
  y = 0.32,
  just = "left",
  gp = gpar(fontsize = 10)
)
```




```
#Export/Save as image -> 1122x590
```

#Temporal distribution (Figure 8)

```
#calculate shares of collisions per month by region, compute shares
y.19.df.b.m <- y.19.df.b %>%
  group_by(regio7bez, UMONAT) %>%
  summarise(count = n()) %>%
  mutate(share = count / sum(count) * 100)

#visualization for month (warning message can be ignored)
m<-ggplot(y.19.df.b.m,aes(UMONAT, share,fill=regio7bez)) +
  geom_bar(stat = "identity",width = 0.7, position = "dodge") +
  theme_bw() +
  ggtitle("Month") +
  scale_x_discrete(
    labels = c(
      "Jan",
      "Feb",
      "Mar",
      "Apr",
      "May",
      "Jun",
      "Jul",
      "Aug",
      "Sep",
      "Oct",
```

```
      "Nov",
      "Dez"
    )
  ) +
  scale_fill_manual(
    "legend",
    values = c(
      "#B42A5D",
      "#FE2F7C",
      "#FF7776",
      "#FFE0DC",
      "#00858B",
      "#41B8B9",
      "#D2E38C"
    ),
    labels = c(
      "Metropolis",
      "Regiopolis",
      "Medium-sized City (u)",
      "Small-town Area (u)",
      "Central City",
      "Medium-sized City (r)",
      "Small-town Area (r)"
    )
  ) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.x = element_text(size = 9),
    axis.text.y = element_text(size = 9),
    legend.position = "bottom",
    legend.title = element_blank()
  )

#calculate shares of collisions per weekday by region, compute shares
y.19.df.b.w <- y.19.df.b %>%
  group_by(regio7bez, UWOCHENTAG) %>%
  summarise(count = n()) %>%
  mutate(share = count / sum(count) * 100)

#visualization for weekdays
w<-ggplot(y.19.df.b.w,
            aes(UWOCHENTAG, share, fill = regio7bez)) +
  geom_bar(stat = "identity",width = 0.7, position= "dodge") +
  theme_bw() +
  ggtitle("Weekdays") +
  scale_x_discrete(
    limits = c(2, 3, 4, 5, 6, 7, 1),
    labels = c("Tue",
               "Wed",
               "Thu",
               "Fri",
```

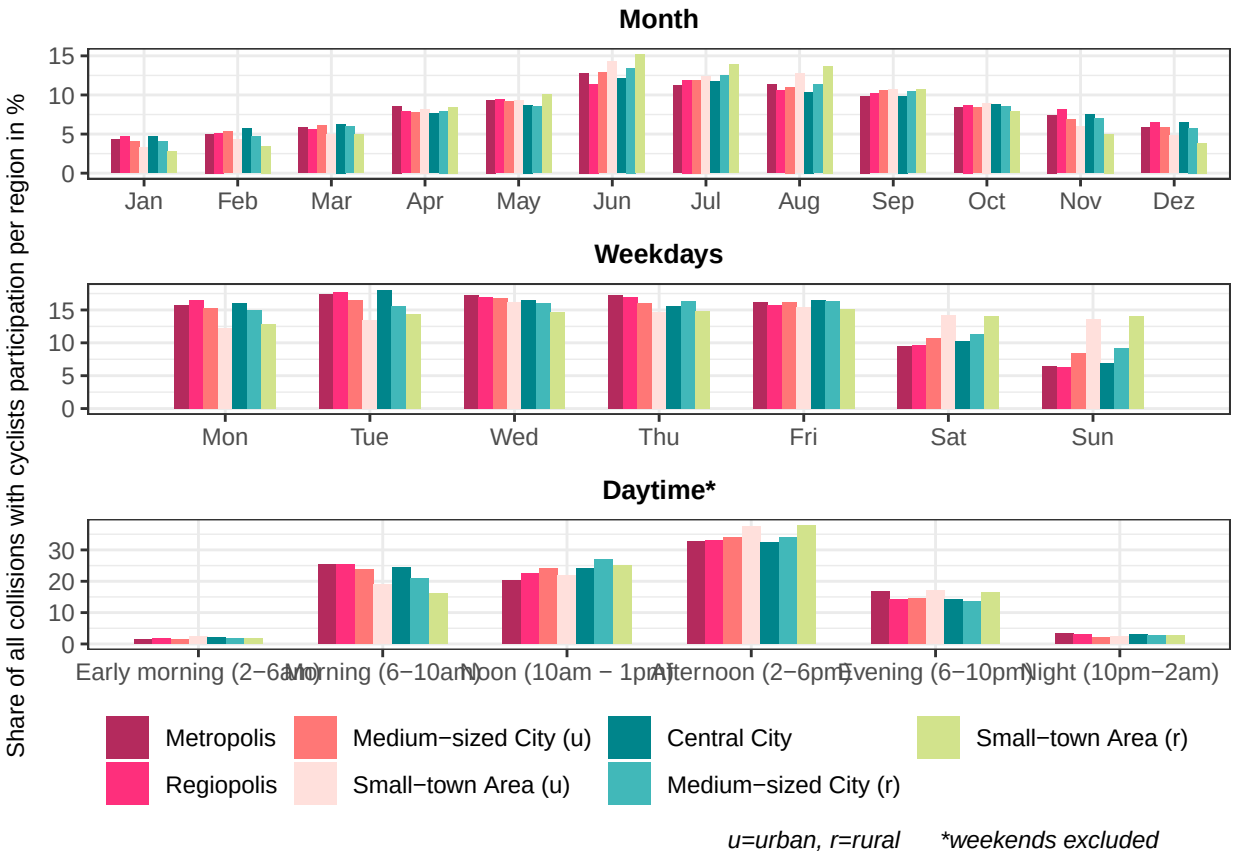
```
                "Sat",
                "Sun",
                "Mon")
  ) +
  scale_fill_manual(
    "legend",
    values = c(
      "#B42A5D",
      "#FE2F7C",
      "#FF7776",
      "#FFE0DC",
      "#00858B",
      "#41B8B9",
      "#D2E38C"
    ),
    labels = c(
      "Metropolis",
      "Regiopolis",
      "Medium-sized City (u)",
      "Small-town Area (u)",
      "Central City",
      "Medium-sized City (r)",
      "Small-town Area (r)"
    )
  ) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.x = element_text(size = 9),
    axis.text.y = element_text(size = 9),
    legend.position = "bottom",
    legend.title = element_blank()
  )
```

```
## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

```
#extract only working days for the hourly shares of collision frequency
y.19.df.b.working <-
  y.19.df.b[!(y.19.df.b$UWOCHENTAG == 7 | y.19.df.b$UWOCHENTAG == 1),]

#calculate shares of collisions per daytime by region, compute shares
y.19.df.b.h <- y.19.df.b.working %>%
  mutate(
    #revalue to daytime categories
    hourcat = case_when(
      USTUNDE %in% c('02', '03', '04', '05') ~ 'night_m',
      USTUNDE %in% c('06', '07', '08', '09') ~ 'morning',
      USTUNDE %in% c('10', '11', '12', '13') ~ 'noon',
      USTUNDE %in% c('14', '15', '16', '17') ~ 'afternoon',
      USTUNDE %in% c('18', '19', '20', '21') ~ 'evening',
      USTUNDE %in% c('22', '23', '00', '01') ~ 'night'
    )
```

```
  ) %>%
  group_by(regio7bez, hourcat) %>%
  summarise(count = n()) %>%
  mutate(share = count / sum(count) * 100)

#visualization for daytimes
h<-ggplot(y.19.df.b.h,
            aes(hourcat, share, fill= regio7bez)) +
  geom_bar(stat = "identity", width = 0.7, position= "dodge") +
  theme_bw() +
  ggtitle("Daytime*") +
  scale_x_discrete(
    limits = c('night_m',
               'morning',
               'noon',
               'afternoon',
               'evening',
               'night'),
    labels = c(
      "Early morning (2-6am)",
      "Morning (6-10am)",
      "Noon (10am - 1pm)",
      "Afternoon (2-6pm)",
      "Evening (6-10pm)",
      "Night (10pm-2am)"
    )
  ) +
  scale_fill_manual(
    "legend",
    values = c(
      "#B42A5D",
      "#FE2F7C",
      "#FF7776",
      "#FFE0DC",
      "#00858B",
      "#41B8B9",
      "#D2E38C"
    ),
    labels = c(
      "Metropolis",
      "Regiopolis",
      "Medium-sized City (u)",
      "Small-town Area (u)",
      "Central City",
      "Medium-sized City (r)",
      "Small-town Area (r)"
    )
  ) +
  theme(
    plot.title = element_text(hjust = 0.5, size = 10, face = "bold"),
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.x = element_text(size = 9),
```

```
    axis.text.y = element_text(size = 9),
    legend.position = "bottom",
    legend.title = element_blank()
  )

#arrange all plots to one graph (Figure 8 in the paper)
figure <-
  ggarrange(
    m,
    w,
    h,
    ncol = 1,
    nrow = 3,
    common.legend = T,
    legend = "bottom"
  )
annotate_figure(
  figure,
  left = text_grob(
    "Share of all collisions with cyclists participation per region in %",
    rot = 90,
    size = 9
  ),
  bottom = text_grob(
    paste0("u=urban, r=rural      *weekends excluded"),
    hjust = 1.2,
    x = 1,
    face = "italic",
    size = 9
  )
)
```

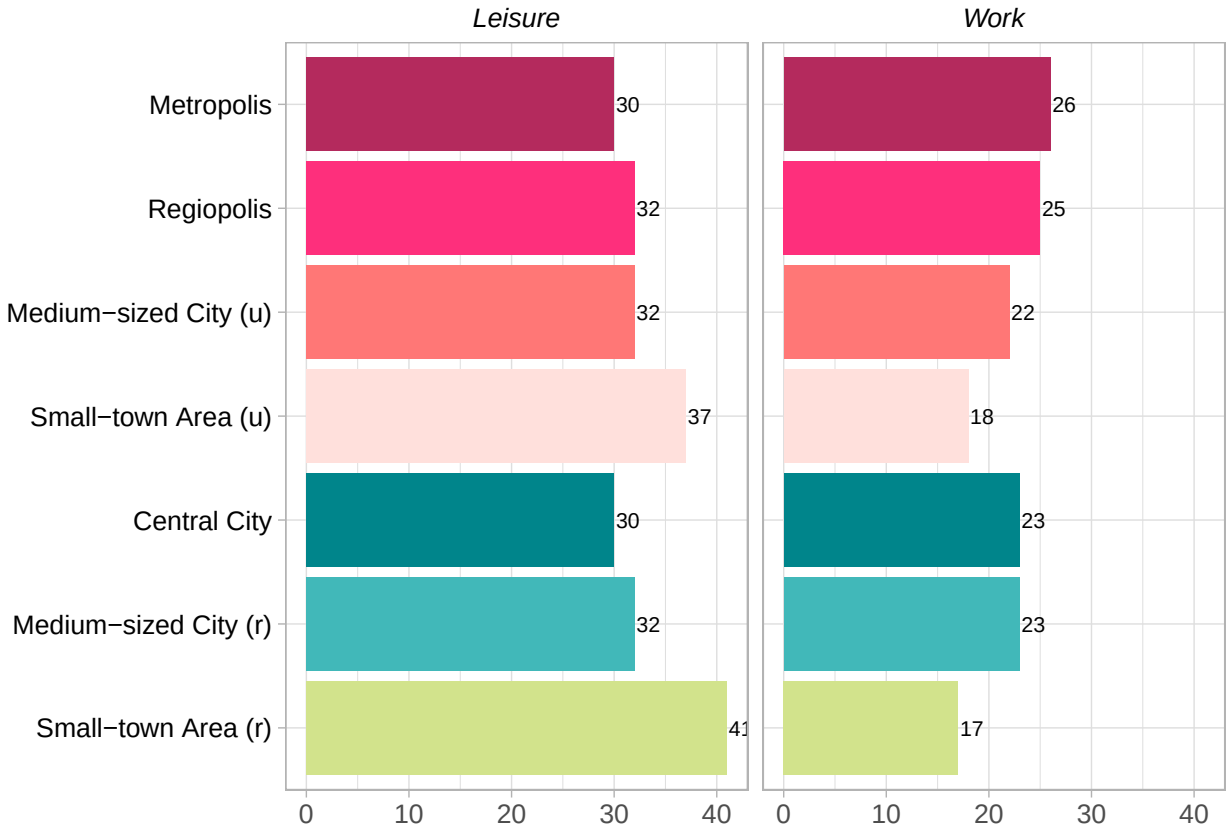



```
#save graph as image
ggsave(
  "month_hour_week22.png",
  width = 8.04,
  height = 6.99,
  dpi = 300
)
```

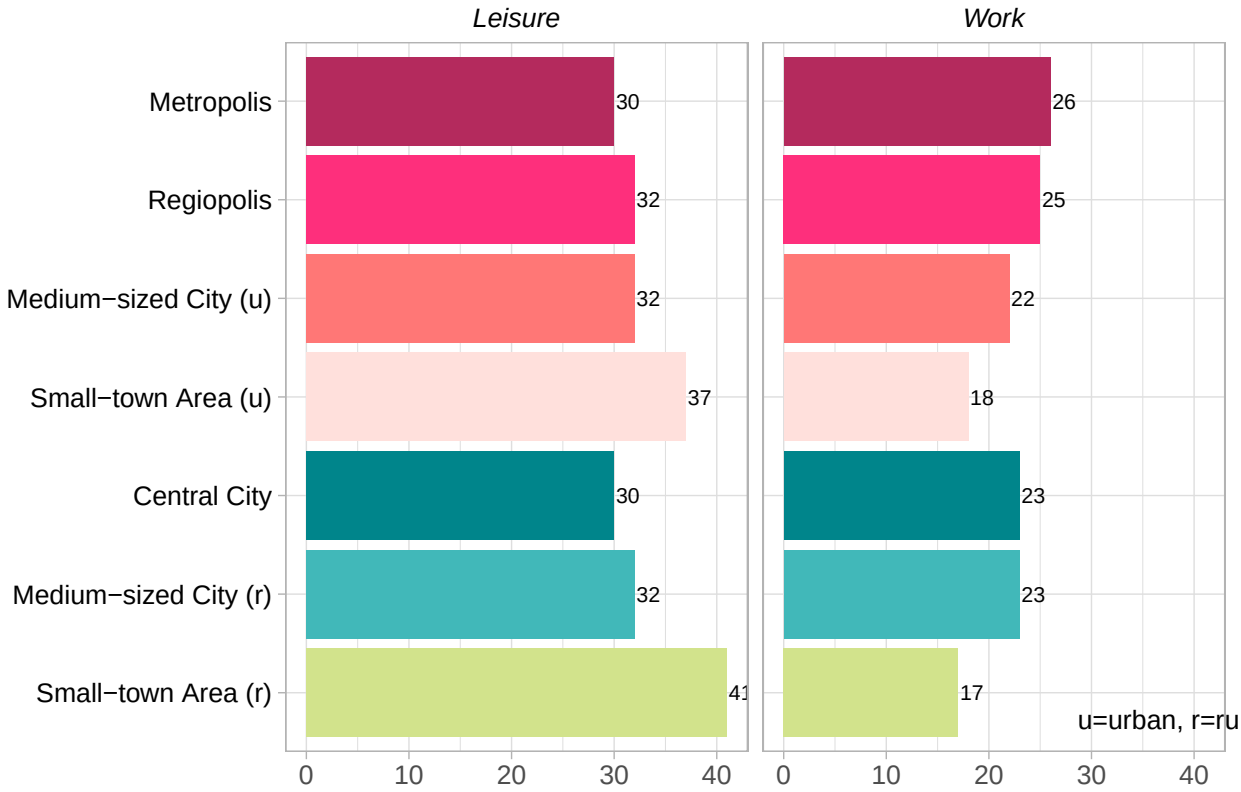
#Trip purpose (Figure 9)

```
#create dataset
regio7bez <-
  c("Metropolis",
    "Regiopolis",
    "Medium-sized City (u)",
    "Small-town Area (u)",
    "Central City",
    "Medium-sized City (r)",
    "Small-town Area (r)"
  )

Work <-
  c(26, 25, 22, 18, 23, 23, 17) #data from Mobilität in Tabellen https://mobilitaet-in-tabellen.dlr.de/
Leisure <-
  c(30, 32, 32, 37, 30, 32, 41) #data from Mobilität in Tabellen https://mobilitaet-in-tabellen.dlr.de/

#create dataframe
```

```r
df.commuting <- data.frame(regio7bez, Work, Leisure)
df.commuting <- df.commuting %>%
  pivot_longer(c(Work, Leisure))

#visualize (Figure 14 in the thesis)
purpose<-ggplot(transform(df.commuting,
                  regio7bez = factor(
                    regio7bez,
                    levels =
                      c(
                        "Small-town Area (r)",
                        "Medium-sized City (r)",
                        "Central City",
                        "Small-town Area (u)",
                        "Medium-sized City (u)",
                        "Regiopolis",
                        "Metropolis"
                      )
                  )),
       aes(regio7bez, value)) +
  theme_light() +
  geom_bar(stat = "identity",width = 0.9, fill = rep(
    c(
      "#B42A5D",
      "#FE2F7C",
      "#FF7776",
      "#FFE0DC",
      "#00858B",
      "#41B8B9",
      "#D2E38C"
    ),
    2
  )) +
  coord_flip() +
  facet_wrap( ~ name, ncol = 2)  +
  geom_text(aes(label = round(value, 1)), hjust = -0.08, size = 2.7) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.x = element_text(size = 10),
    axis.text.y = element_text(color = 'black', size = 10),
    strip.text = element_text(color = 'black', size = 10, face = "italic"),
    strip.background = element_blank()
  )

#add footnote to ggplot graph
print(purpose)
```




```
annotate_figure(purpose,
  bottom = text_grob(
    paste0("u=urban, r=rural      *weekends excluded"),
  x = 0.87,
  y = 3.22,
  just = "left",
  size = 10)
)
```




```
#Export/Save as image -> 860x404
```