R Lab4: Missing Data and Graph

Haocheng Hu and Zhaopeng Qu

Nanjing University

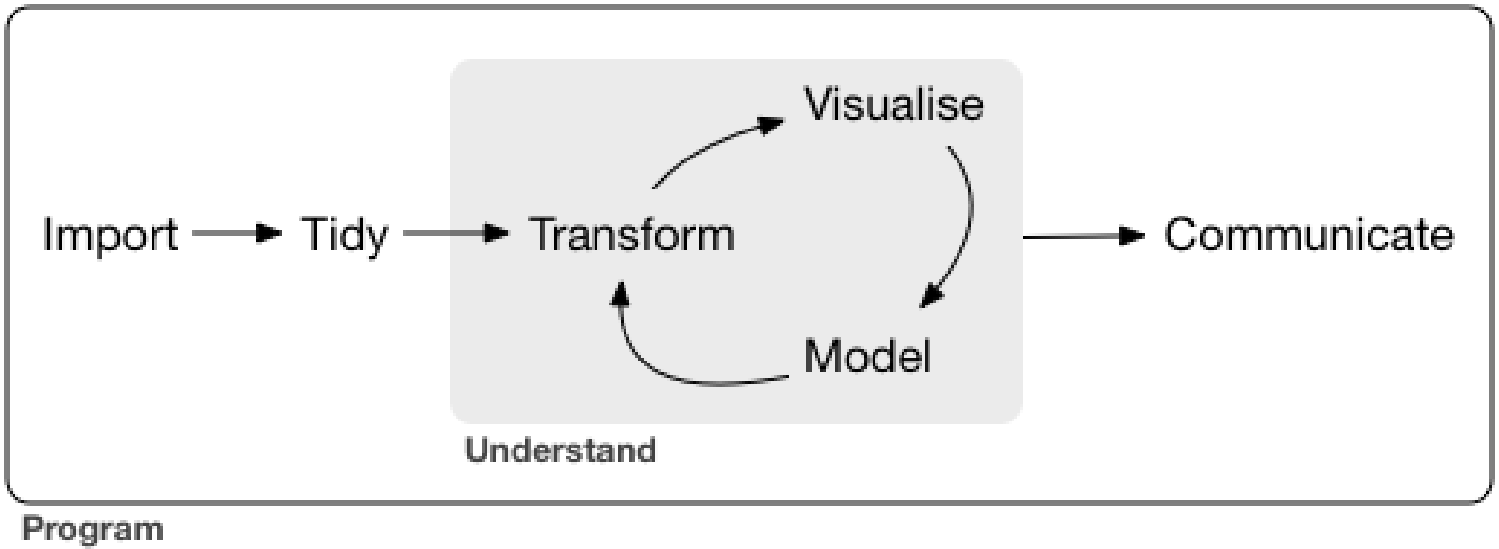
3/19/2025

Section 1

Review Last Lecture

Review: Tidy Data and the Workflow

- Variable and Data Types in R
- Data Structure in R
- Two Models to clean data in R
  - Basic R
  - `Tidyverse`
- The **workflow** of data analysis is as follows



Section 2

Introduction to Missing values

Introduction

- Working with *real-world* data = *working with missing data*.
- Missing values are values that *should have been recorded but were not*.
- Missing values in R
  - Explicitly: **NA** = *Not Available* or other forms
  - Implicitly: simply not present in the data

Missing values in Data

- Find and unify missing values
- Summarize missing values
- Deal with missing values

Find missing values

- Whether easy to tell
  - explicit
  - implicit(panel)
- Whether united forms

A Special Data Case

```
df <- read_csv("Data/telecom.csv")
glimpse(df)
```

```
## Rows: 10
## Columns: 5
## $ customerID     <chr> "7590-VHVEG", "5575-GNVDE", "3668-QPYBK", "7795-C
## $ MonthlyCharges <dbl> 29.85, 56.95, NA, 42.30, 70.70, NaN, 89.10, NA, 1
## $ TotalCharges   <chr> "109.9", "na", "108.15", "1840.75", NA, "820.5",
## $ PaymentMethod  <chr> "Electronic check", "Mailed check", "--", "Bank t
## $ Churn          <chr> "yes", "yes", "yes", "no", "no", "yes", "no", "ye
```

Find missing value commands

- `is.na()`
- `any_na()`
- `n_miss_row()`

Missing values in Data

```
df$MonthlyCharges
```

```
##   [1]   29.85   56.95      NA   42.30   70.70     NaN   89.10      N
```

```
is.na(df$MonthlyCharges)
```

```
##   [1] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE
```

- R only identified "NA" as a missing value.

```
is.nan(df$MonthlyCharges)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
```

- Missing values in R
  - **NaN** = *Not a Number*

Missing Values in Data: More

- “Missing”
- “-99999”
- “N/A”
- “.”
- “”
- ” ”
- ...
- “.d” : did not know
- “.e” : inappropriate
- “.r” : refused

CASE: Missing Values

```
head(df)
```

```
## # A tibble: 6 x 5
##   customerID MonthlyCharges TotalCharges PaymentMethod    C
##   <chr>                <dbl> <chr>        <chr>            <
## 1 7590-VHVEG            29.8 109.9        Electronic check y
## 2 5575-GNVDE            57.0 na           Mailed check     y
## 3 3668-QPYBK             NA  108.15       --               y
## 4 7795-CFOCW            42.3 1840.75      Bank transfer    n
## 5 9237-HQITU            70.7 <NA>         Electronic check n
## 6 9305-CDSKC           NaN   820.5        --               y
```

Unified forms of missing values

```
#install.packages("naniar")
```

- `naniar:: replace_with_na`: turn missing values into NA
  - `replace_with_na`
  - `replace_with_na_all()`
  - `replace_with_na_at()`
  - `replace_with_na_if()`
- `tidyr::replace_na`: turn missing values(NA) into a value

Replacing missing values with “NA”

```
df$PaymentMethod
```

```
##   [1] "Electronic check" "Mailed check"      "--"
##   [5] "Electronic check" "--"                "Credit card"
##   [9] "Electronic check" "Electronic check"
```

Missing value in R: Replacing missing values

```
df %>%
replace_with_na(replace = list(PaymentMethod=c("","--")))
```

```
## # A tibble: 10 x 5
##    customerID MonthlyCharges TotalCharges PaymentMethod
##    <chr>                <dbl> <chr>        <chr>
##  1 7590-VHVEG            29.8 109.9        Electronic check
##  2 5575-GNVDE            57.0 na           Mailed check
##  3 3668-QPYBK             NA  108.15       <NA>
##  4 7795-CFOCW            42.3 1840.75       Bank transfer
##  5 9237-HQITU            70.7 <NA>         Electronic check
##  6 9305-CDSKC           NaN   820.5        <NA>
##  7 1452-KIOVK            89.1 1949.4        Credit card
##  8 6713-OKOMC             NA  N/A          <NA>
##  9 7892-POOKP           105.  3046.05       Electronic check
## 10 8451-AJOMK            54.1 354.95        Electronic check
```

Replacing missing values

- Practice by your self
  - replace all potential missing values with “NA”

Analysis Missing Values

- Before dealing with missing values in the data, it's important to find them and figure out why they exist in the first place
  - **MCAR**: Missing Completely at Random
  - **MAR**: Missing At Random
  - **MNAR**: Missing Not At Random

Analysis Missing Values: MCAR

Missingness has no association with any data you have observed, or not observed.

test	vacation
NA	TRUE
11.533340	FALSE
10.126115	TRUE
NA	FALSE
NA	TRUE
8.551881	FALSE
NA	FALSE
NA	TRUE
10.608264	TRUE
8.611877	TRUE

Analysis Missing Values: MAR

Missingness depends on data observed, but not data observed

Implications:

- Impute
- Deleting observations not ideal, may lead to bias

test	vacation	depression
NA	TRUE	87.93109
11.533340	FALSE	40.02708
10.126115	TRUE	48.62883
NA	FALSE	88.21743
NA	TRUE	90.29282
8.551881	FALSE	44.77343
NA	FALSE	89.48865
NA	TRUE	89.99209
10.608264	TRUE	45.56832
8.611877	TRUE	42.41686

Analysis Missing Values: MNAR

Missingness of the response is related to an unobserved value relevant to the assessment of interest.

Implications:

- Data will be biased from deletion and imputation
- Inference can be limited, proceed with caution.

test	vacation	depression
NA	TRUE	NA
11.533340	FALSE	11.533340
10.126115	TRUE	10.126115
NA	FALSE	NA
NA	TRUE	NA
8.551881	FALSE	8.551881
NA	FALSE	NA
NA	TRUE	NA
10.608264	TRUE	10.608264
8.611877	TRUE	8.611877

Summary of missing values

- R Function
  - `n_miss`
  - `n_complete`
  - `miss_var_summary`
  - `miss_case_summary`

Summary of missing values in R

```
airquality
```

```
##     Ozone Solar.R Wind Temp Month Day
## 1      41     190  7.4   67     5   1
## 2      36     118  8.0   72     5   2
## 3      12     149 12.6   74     5   3
## 4      18     313 11.5   62     5   4
## 5      NA      NA 14.3   56     5   5
## 6      28      NA 14.9   66     5   6
## 7      23     299  8.6   65     5   7
## 8      19      99 13.8   59     5   8
## 9       8      19 20.1   61     5   9
## 10     NA     194  8.6   69     5  10
## 11      7      NA  6.9   74     5  11
## 12     16     256  9.7   69     5  12
## 13     11     290  9.2   66     5  13
```

Summary of missing values in R

```
n_complete(airquality)
```

```
## [1] 874
```

```
n_miss(airquality)
```

```
## [1] 44
```

Summary of missing values in R

```
miss_var_summary(airquality)
```

```
## # A tibble: 6 x 3
##   variable n_miss pct_miss
##   <chr>     <int>    <num>
## 1 Ozone        37    24.2
## 2 Solar.R       7     4.58
## 3 Wind          0     0
## 4 Temp          0     0
## 5 Month         0     0
## 6 Day           0     0
```

Summary of missing values in R

```
miss_case_summary(airquality)
```

```
## # A tibble: 153 x 3
##     case n_miss pct_miss
##    <int>  <int>    <dbl>
##  1     5      2     33.3
##  2    27      2     33.3
##  3     6      1     16.7
##  4    10      1     16.7
##  5    11      1     16.7
##  6    25      1     16.7
##  7    26      1     16.7
##  8    32      1     16.7
##  9    33      1     16.7
## 10    34      1     16.7
## # i 143 more rows
```

Summary of missing values in R

```
miss_var_table(airquality)
```

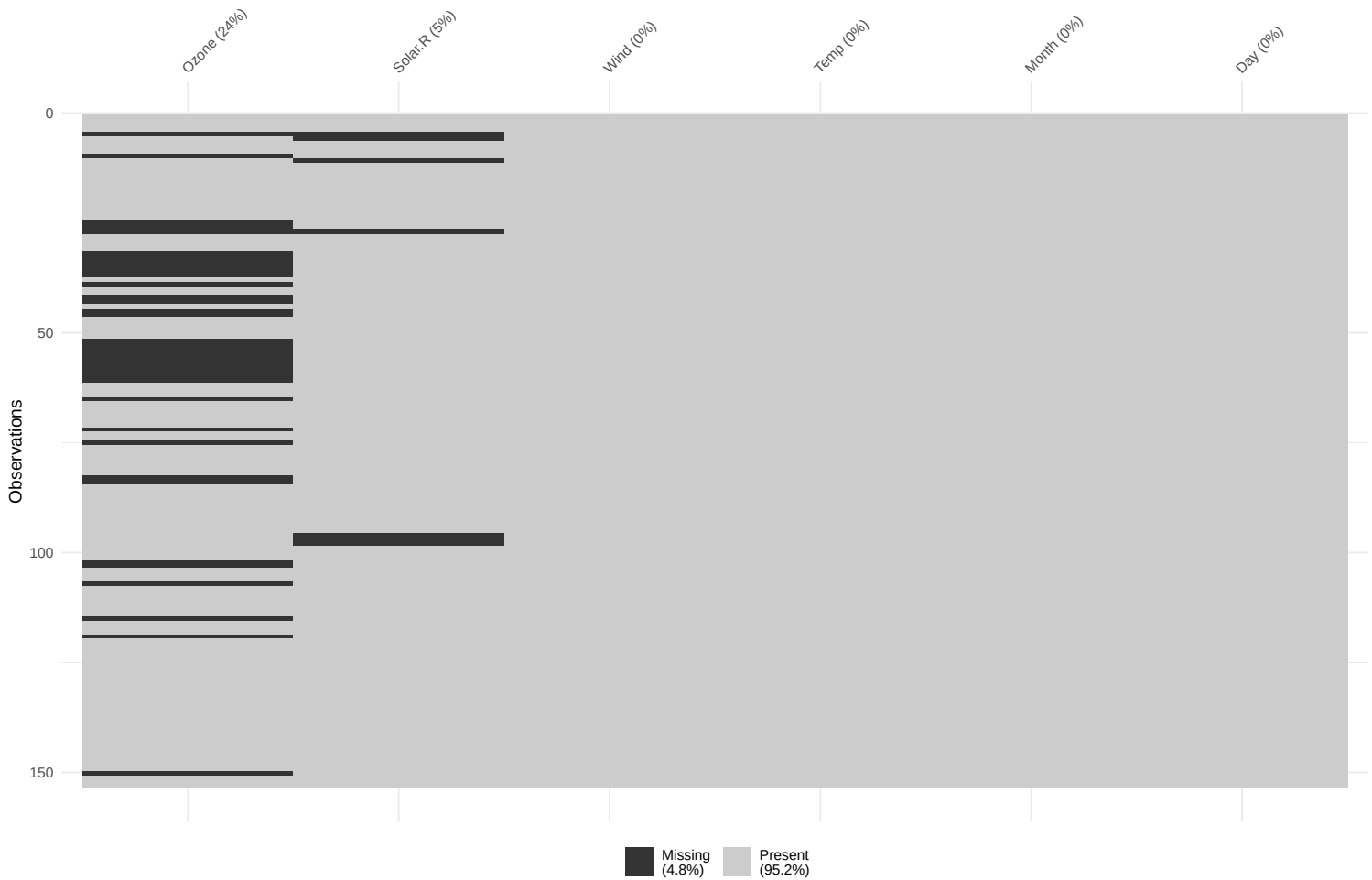
```
## # A tibble: 3 x 3
##   n_miss_in_var n_vars pct_vars
##           <int>  <int>    <dbl>
## 1             0      4     66.7
## 2             7      1     16.7
## 3            37      1     16.7
```

```
miss_case_table(airquality)
```

```
## # A tibble: 3 x 3
##   n_miss_in_case n_cases pct_cases
##            <int>   <int>     <dbl>
## 1              0     111      72.5
## 2              1      40      26.1
```

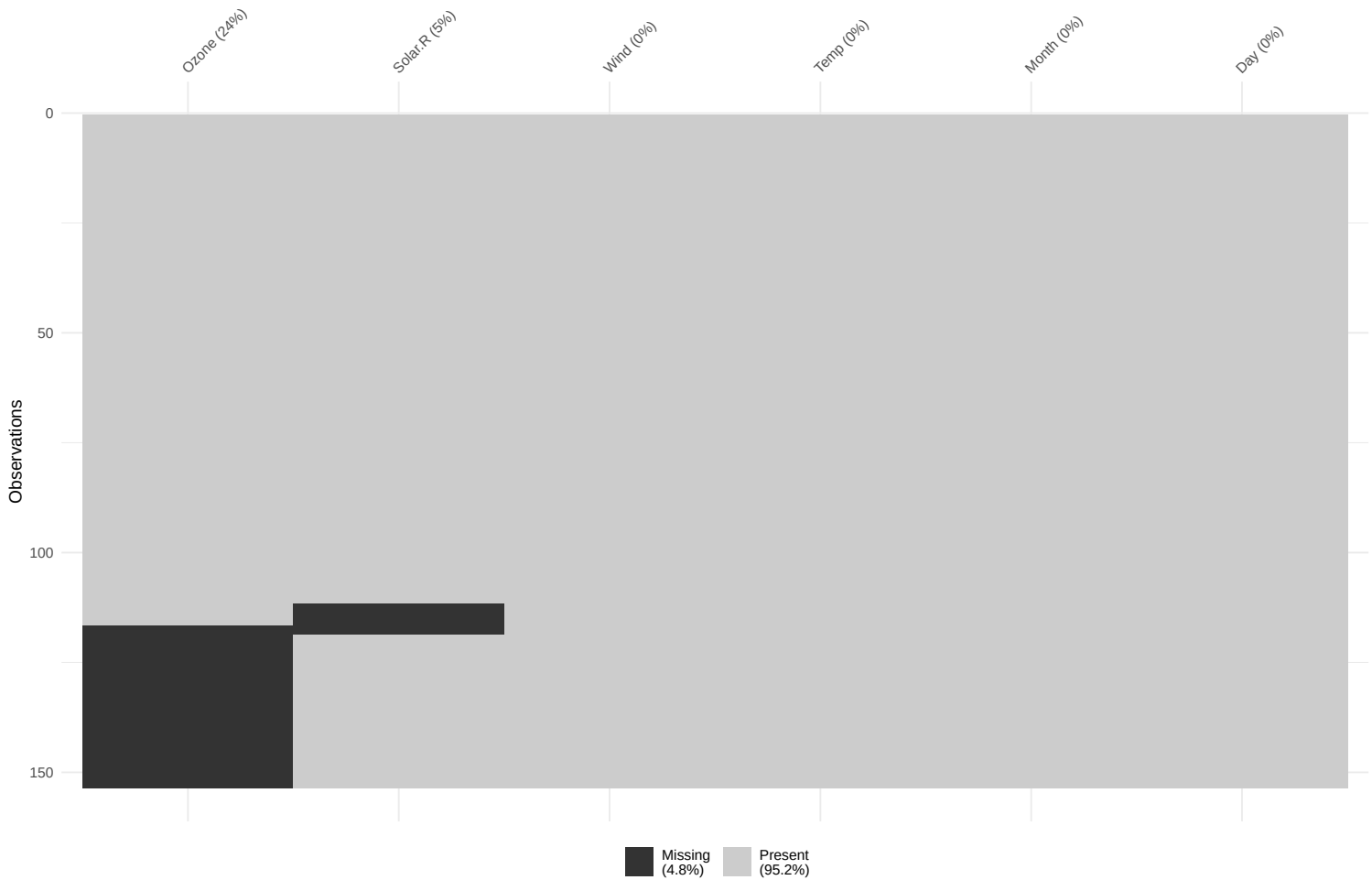
Visualizations of missing values in R

```
vis_miss(airquality)
```



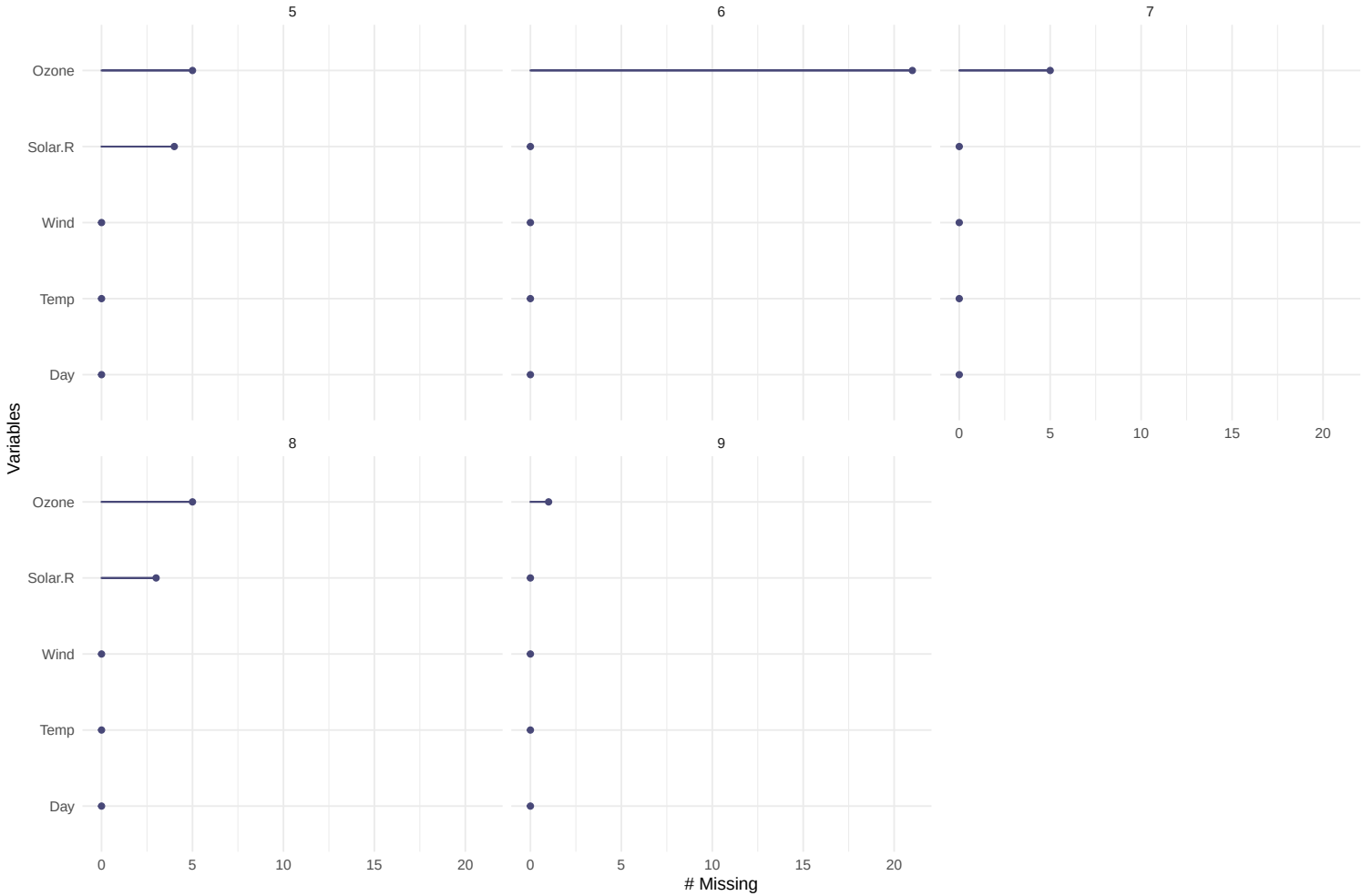
Visualizations of missing values in R

```
vis_miss(airquality,cluster = T)
```



Visualizations of missing values in R

```
gg_miss_var(airquality, facet = Month)
```



Dealing with Missing Values in practice

- No perfect way to deal with. It has to be case by case.
- Three ways to deal with
  - deleting the observations
  - removing the variable
  - imputating

Missing Values: Deleting the observations.

- The share of the missing observations is **minor**,let's say below 5%.
- or **"MCAR"** missing
- Then it can be dropped directly

```
df %>% drop_na()
```

```
## # A tibble: 6 x 5
##   customerID MonthlyCharges TotalCharges PaymentMethod    C
##   <chr>                <dbl> <chr>        <chr>            <
## 1 7590-VHVEG            29.8 109.9        Electronic check y
## 2 5575-GNVDE            57.0 na           Mailed check     y
## 3 7795-CFOCW            42.3 1840.75      Bank transfer    n
## 4 1452-KIOVK            89.1 1949.4       Credit card      n
## 5 7892-POOKP           105.  3046.05      Electronic check n
## 6 8451-AJOMK            54.1 354.95       Electronic check n
```

Missing Values: imputating

- What if
  1. deleting observations will leads a small sample size.
  2. the share of the missing observations is over 20%.
- imputating with statistics
  - mean,median,mode
  - the next(previous) value
  - regression prediction or matching

Missing Values: imputating

- `tidyr::replace_na`: turn missing values(NAs) into specific values

```
df$PaymentMethod
```

```
##  [1] "Electronic check" "Mailed check"     "--"
##  [5] "Electronic check" "--"               "Credit card"
##  [9] "Electronic check" "Electronic check"
```

Replacing missing values

```
df %>%
replace_with_na(replace = list(PaymentMethod=c("","--"))) %>%
replace_na(list(PaymentMethod="Unknown"))
```

```
## # A tibble: 10 x 5
##    customerID MonthlyCharges TotalCharges PaymentMethod    Churn
##    <chr>                <dbl> <chr>        <chr>            <chr>
##  1 7590-VHVEG            29.8 109.9        Electronic check yes
##  2 5575-GNVDE            57.0 na           Mailed check     yes
##  3 3668-QPYBK            NA   108.15       Unknown          yes
##  4 7795-CFOCW            42.3 1840.75      Bank transfer    no
##  5 9237-HQITU            70.7 <NA>         Electronic check no
##  6 9305-CDSKC           NaN   820.5        Unknown          yes
##  7 1452-KIOVK            89.1 1949.4       Credit card      no
##  8 6713-OKOMC            NA   N/A          Unknown          yes
##  9 7892-POOKP           105.  3046.05      Electronic check no
## 10 8451-AJOMK            54.1 354.95       Electronic check no
```

Replacing missing values with statistics

```
df %>%
  mutate(MonthlyCharges=
           replace(MonthlyCharges,is.na(MonthlyCharges),
                   mean(MonthlyCharges, na.rm = TRUE)))
```

```
## # A tibble: 10 x 5
##    customerID MonthlyCharges TotalCharges PaymentMethod    Churn
##    <chr>                <dbl> <chr>        <chr>            <chr>
##  1 7590-VHVEG            29.8 109.9        Electronic check yes
##  2 5575-GNVDE            57.0 na           Mailed check     yes
##  3 3668-QPYBK            64.0 108.15       --               yes
##  4 7795-CFOCW            42.3 1840.75      Bank transfer    no
##  5 9237-HQITU            70.7 <NA>         Electronic check no
##  6 9305-CDSKC            64.0 820.5        --               yes
##  7 1452-KIOVK            89.1 1949.4       Credit card      no
##  8 6713-OKOMC            64.0 N/A          <NA>             yes
##  9 7892-POOKP           105.  3046.05       Electronic check no
## 10 8451-AJOMK            54.1 354.95       Electronic check no
```

Replacing missing values with statistics

- Exercise: Replacing missing values in `TotalCharges` with median values

Replacing missing values with statistics

- Exercise: Replacing missing values in TotalCharges with median values

```
df %>%
replace_with_na(replace = list(TotalCharges=c("na","N/A"))) %>
  mutate(TotalCharges=as.numeric(TotalCharges)) %>%
  mutate(TotalCharges=replace(TotalCharges,is.na(TotalCharges),
                              mean(TotalCharges,na.rm = TRUE)))
```

```
## # A tibble: 10 x 5
##    customerID MonthlyCharges TotalCharges PaymentMethod
##    <chr>               <dbl>        <dbl> <chr>
##  1 7590-VHVEG           29.8         110. Electronic check
##  2 5575-GNVDE           57.0        1176. Mailed check
##  3 3668-QPYBK           NA           108. --
##  4 7795-CFOCW           42.3        1841. Bank transfer
##  5 9237-HQITU           70.7        1176. Electronic check
```

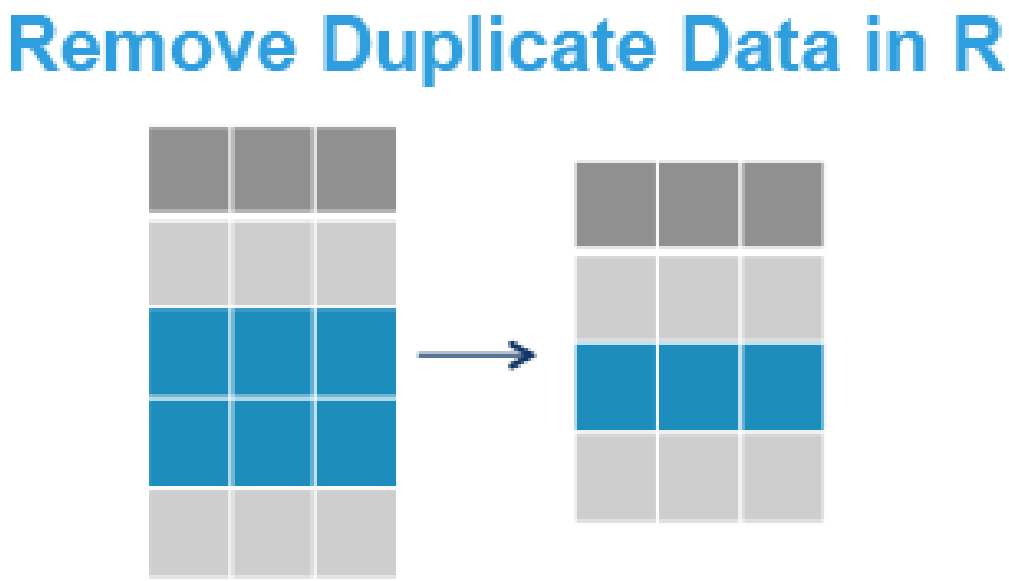
Section 3

Duplicates

What is Duplicates?

- It means that
  - some observations are duplicated (maybe due to replicating records) in *all variables*.
  - some observation are only duplicated *only in one or several variables*.

Packages for Duplicates



duplicated(): *Identify duplicate elements (R base)*
unique(): *Keep only unique elements (R base)*
distinct(): *Efficient solution to remove duplicate in a data table (dplyr)*

Find and drop duplicate elements

```r
x <- c(1, 1, 4, 5, 4, 6)
duplicated(x)
```

```
## [1] FALSE  TRUE FALSE FALSE  TRUE FALSE
```

```r
x[duplicated(x)] # Extract duplicate elements
```

```
## [1] 1 4
```

```r
x[!duplicated(x)] #remove duplicated elements
```

```
## [1] 1 4 5 6
```

Find and drop duplicate elements

```
unique(x)
```

```
## [1] 1 4 5 6
```

Section 4

Data visualization: ggplot2 in Tidyverse

Introduction

- ggplot2 is a powerful data exploration and visualization package that can create graphics in R. It was created by Hadley Wickham who is a leading developer of R package.
- This function is the implementation of the "Grammar of Graphics" that allows us to build layers of graphical elements to produce plots.

```
library(ggplot2)
library(gapminder)
```

```
## Warning: package 'gapminder' was built under R version 4.4.
```

Terminology in ggplot2

- **ggplot** - the main function where you specify the data set and variables to plot (this is where we define the `x` and `y` variable names)
- **geoms** - geometric objects
  - e.g. `geom_point()`, `geom_bar()`, `geom_line()`, `geom_histogram()`,`geom_boxplot()`
- **aes** - aesthetics
  - shape, transparency, color, fill, linetype
- **scales** - define how your data will be plotted
  - continuous, discrete, log, etc

Scatter Plot

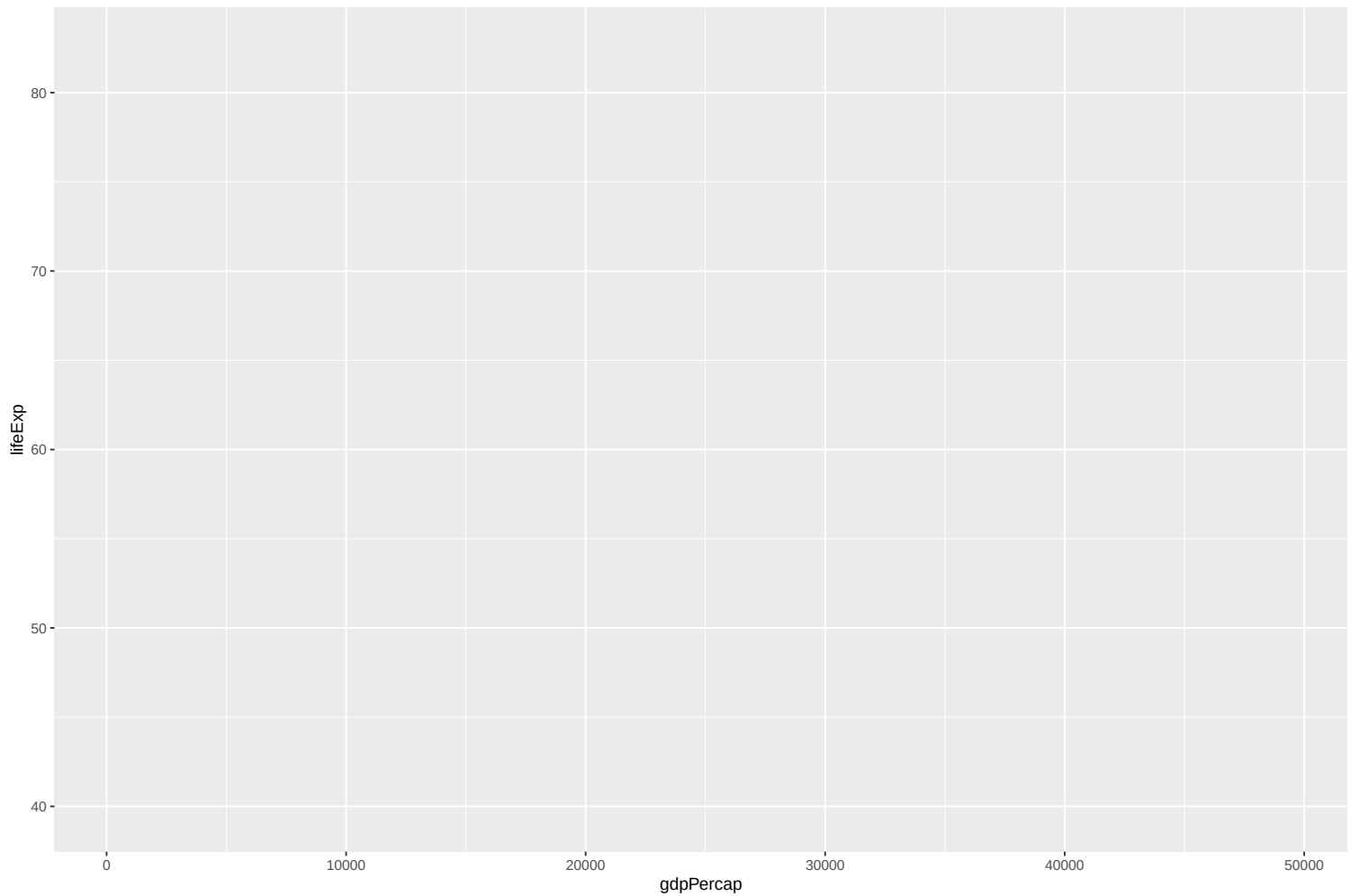
- generate a subset data in 2007

```
gapminder_2007 <- gapminder %>%
    filter(year==2007)
glimpse(gapminder_2007)
```

```
## Rows: 142
## Columns: 6
## $ country   <fct> "Afghanistan", "Albania", "Algeria", "Ang
## $ continent <fct> Asia, Europe, Africa, Africa, Americas, O
## $ year      <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007,
## $ lifeExp   <dbl> 43.828, 76.423, 72.301, 42.731, 75.320, 8
## $ pop       <int> 31889923, 3600523, 33333216, 12420476, 40
## $ gdpPercap <dbl> 974.5803, 5937.0295, 6223.3675, 4797.2313
```

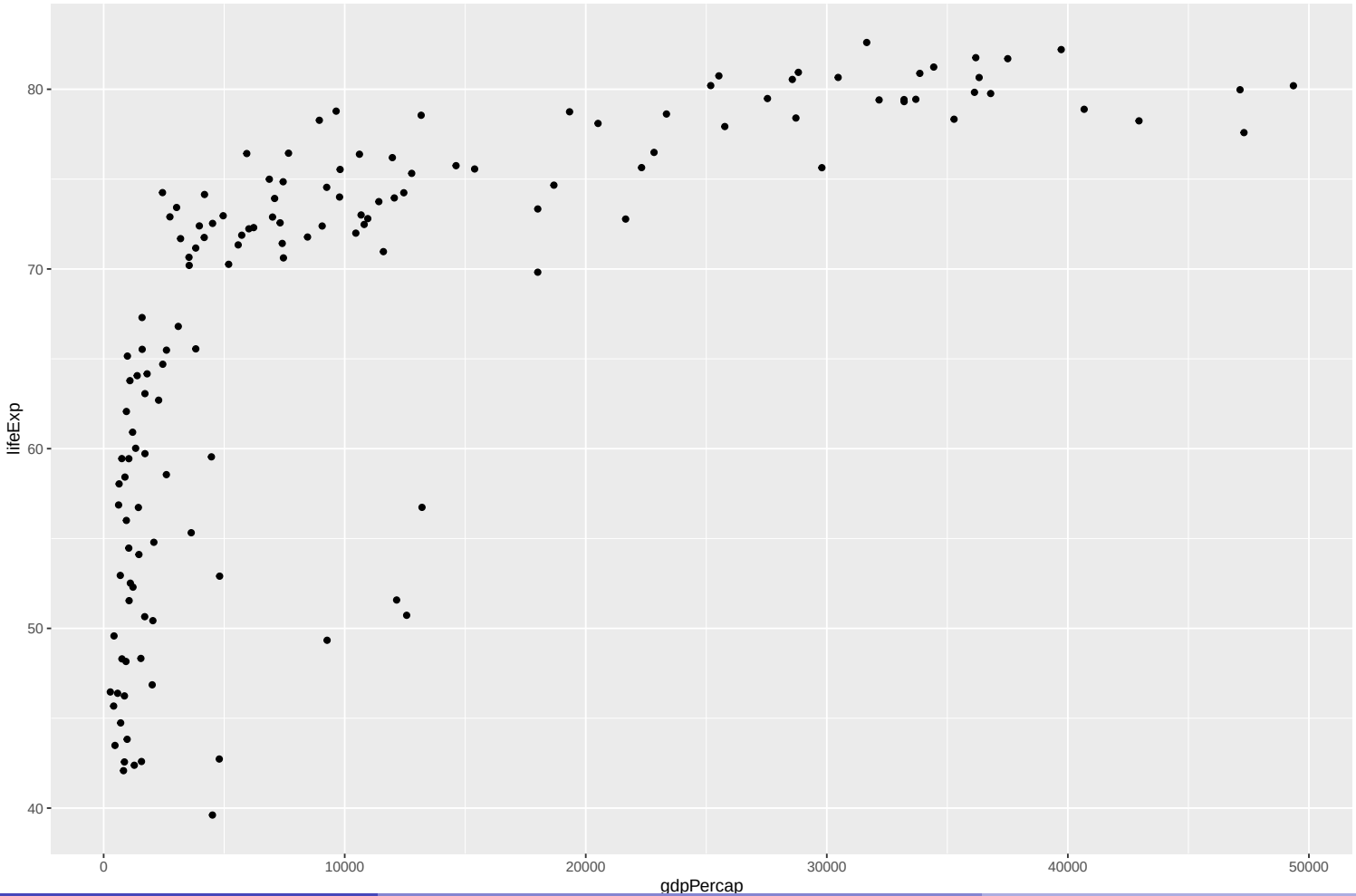
Scatter Plot

```
ggplot(gapminder_2007,aes(x=gdpPercap, y = lifeExp))
```



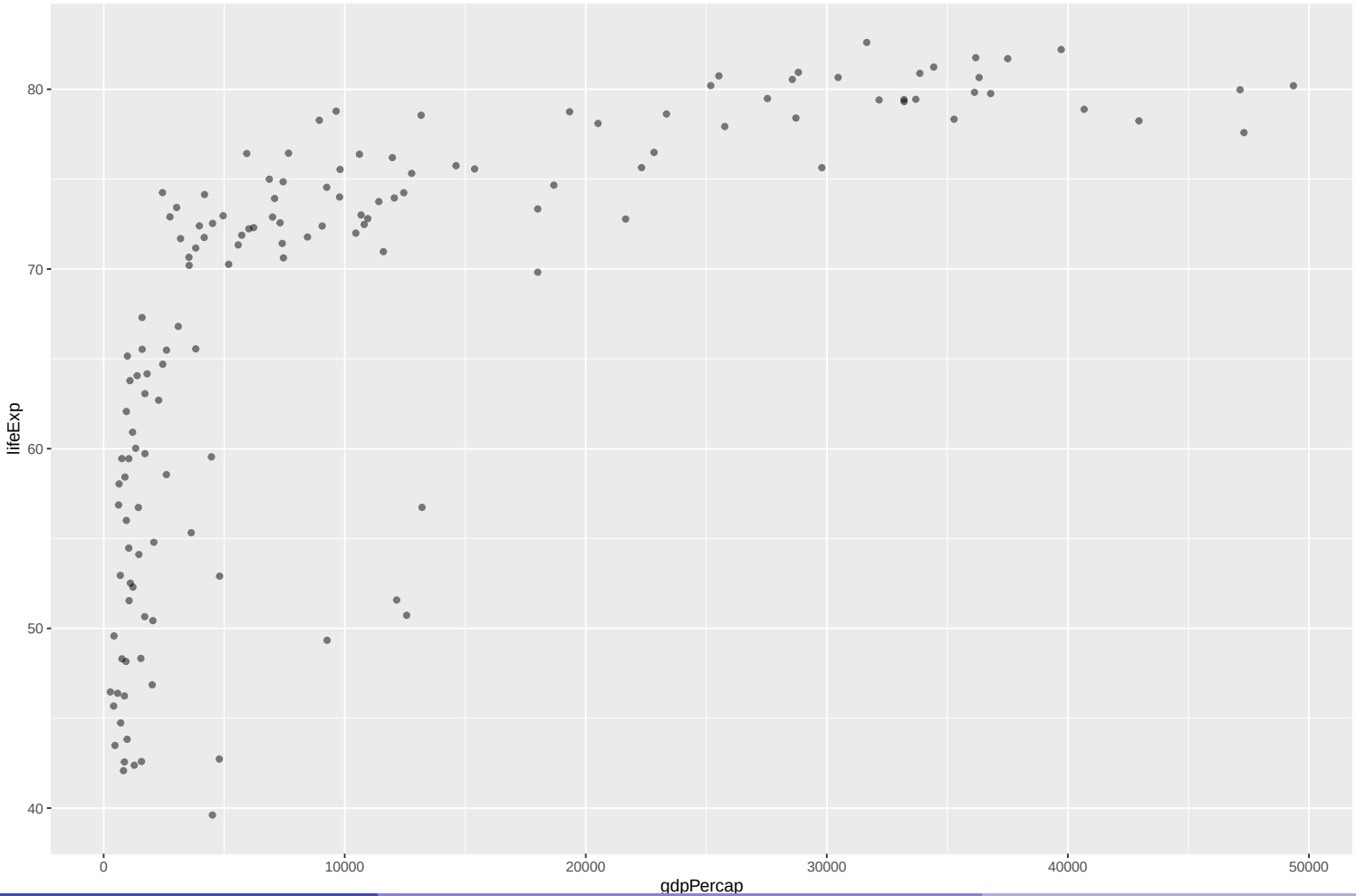
Scatter Plot

```
ggplot(gapminder_2007,aes(x=gdpPercap, y = lifeExp)) +
geom_point()
```



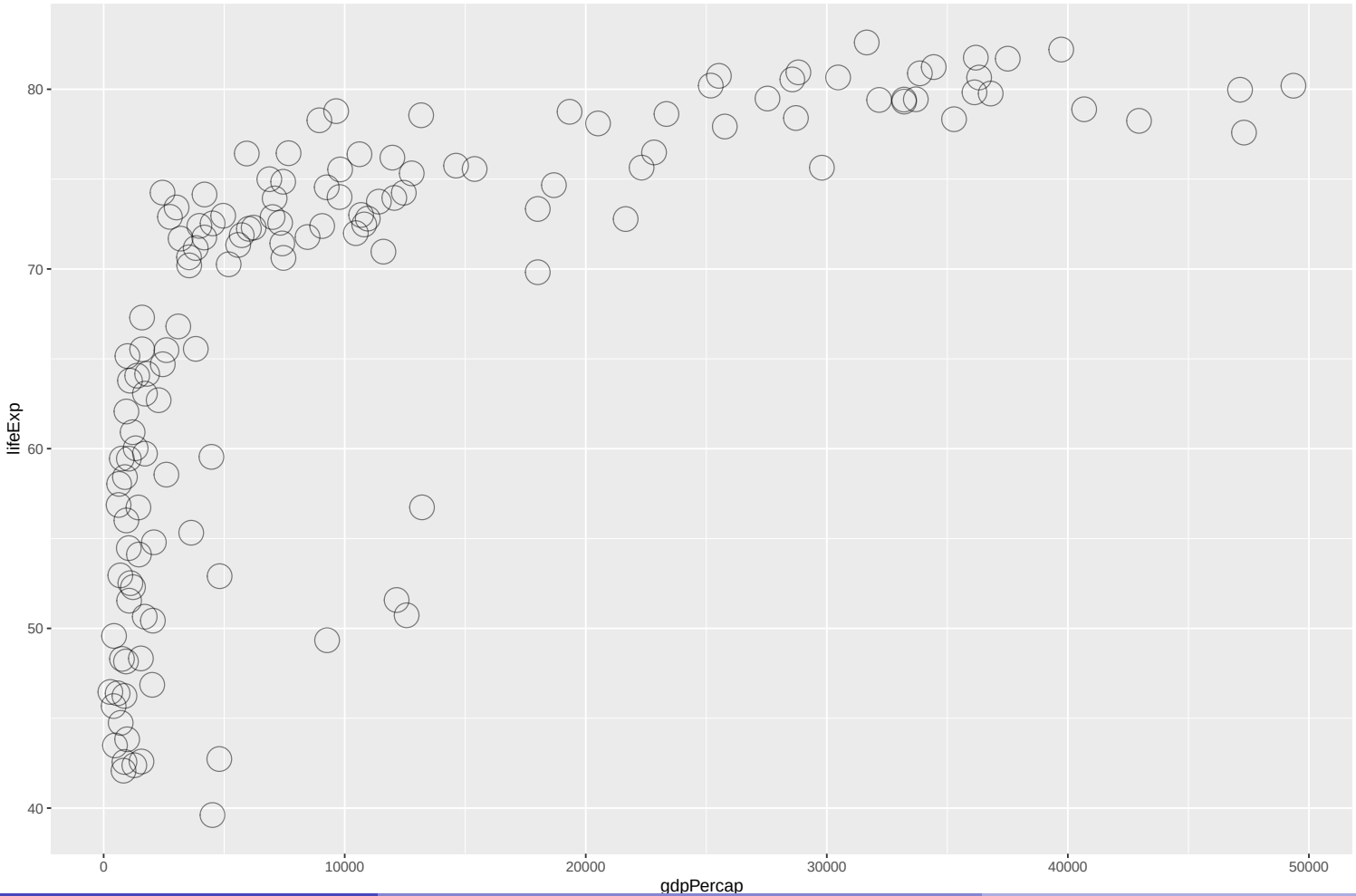
Scatter Plot

```
ggplot(gapminder_2007,aes(x=gdpPercap, y = lifeExp)) +
geom_point(alpha=0.5)
```



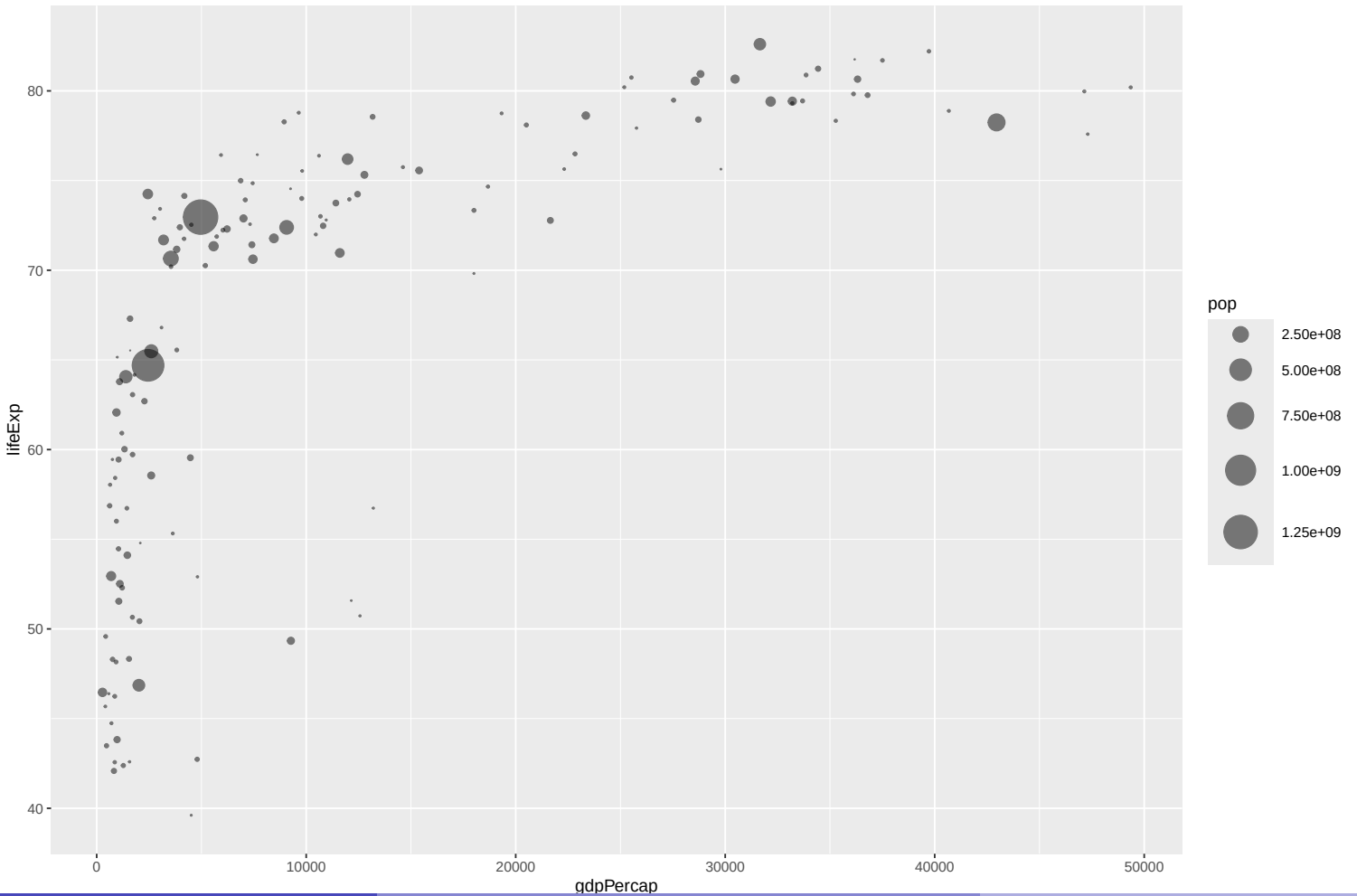
Scatter Plot

```
ggplot(gapminder_2007,aes(x=gdpPercap, y = lifeExp)) +
geom_point(alpha=0.5,shape=21,size=7)
```



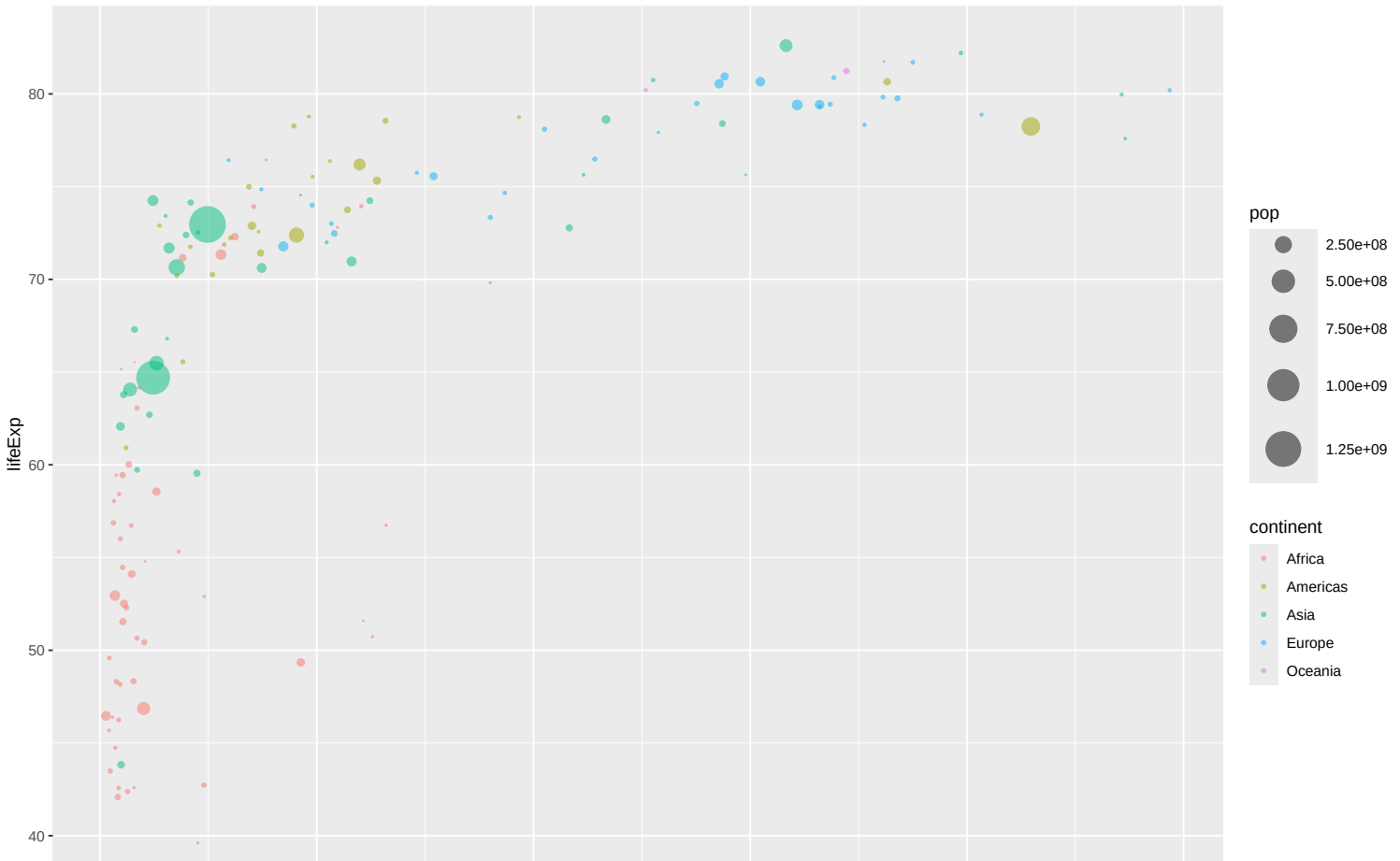
Scatter Plot

```
ggplot(gapminder_2007,aes(x=gdpPercap, y = lifeExp,size=pop)) +
geom_point(alpha=0.5,shape=20)+scale_size(range=c(.1,15))
```



Scatter Plot in colors

```
ggplot(gapminder_2007,
    aes(x=gdpPercap, y = lifeExp,size=pop,color=continent)) +
    geom_point(alpha=0.5,shape=20)+scale_size(range=c(.1,15))
```



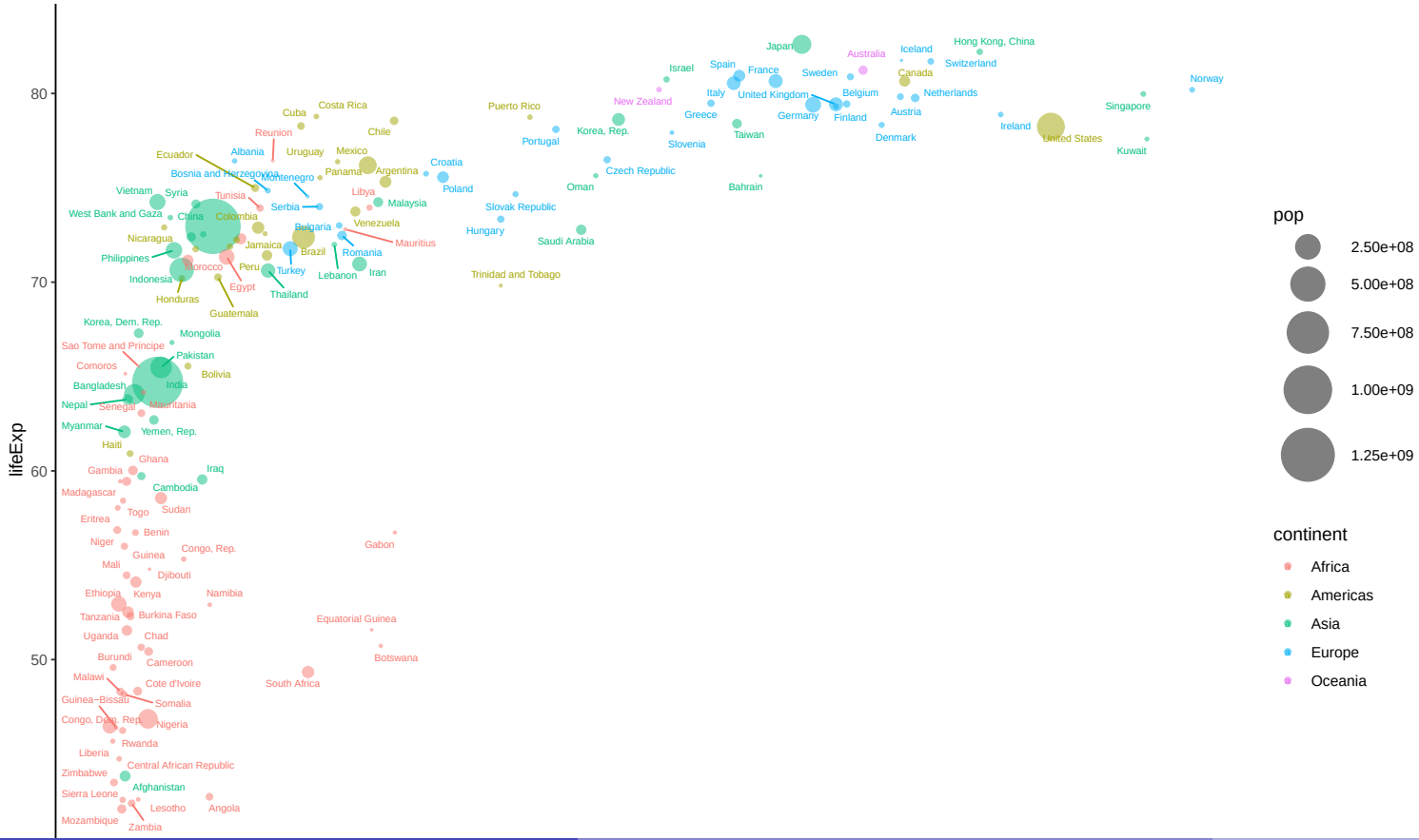
Scatter Plot:label in ggrepel

```
# install.packages("ggrepel")
library(ggrepel)
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(x=gdpPercap,
               y = lifeExp,size = pop,color=continent)) +
    geom_point(alpha=0.5)+scale_size(range=c(.1,15)) +
    geom_text_repel(aes(label = country), size = 2) +
    theme_classic()
```

Scatter Plot:label in ggrepel

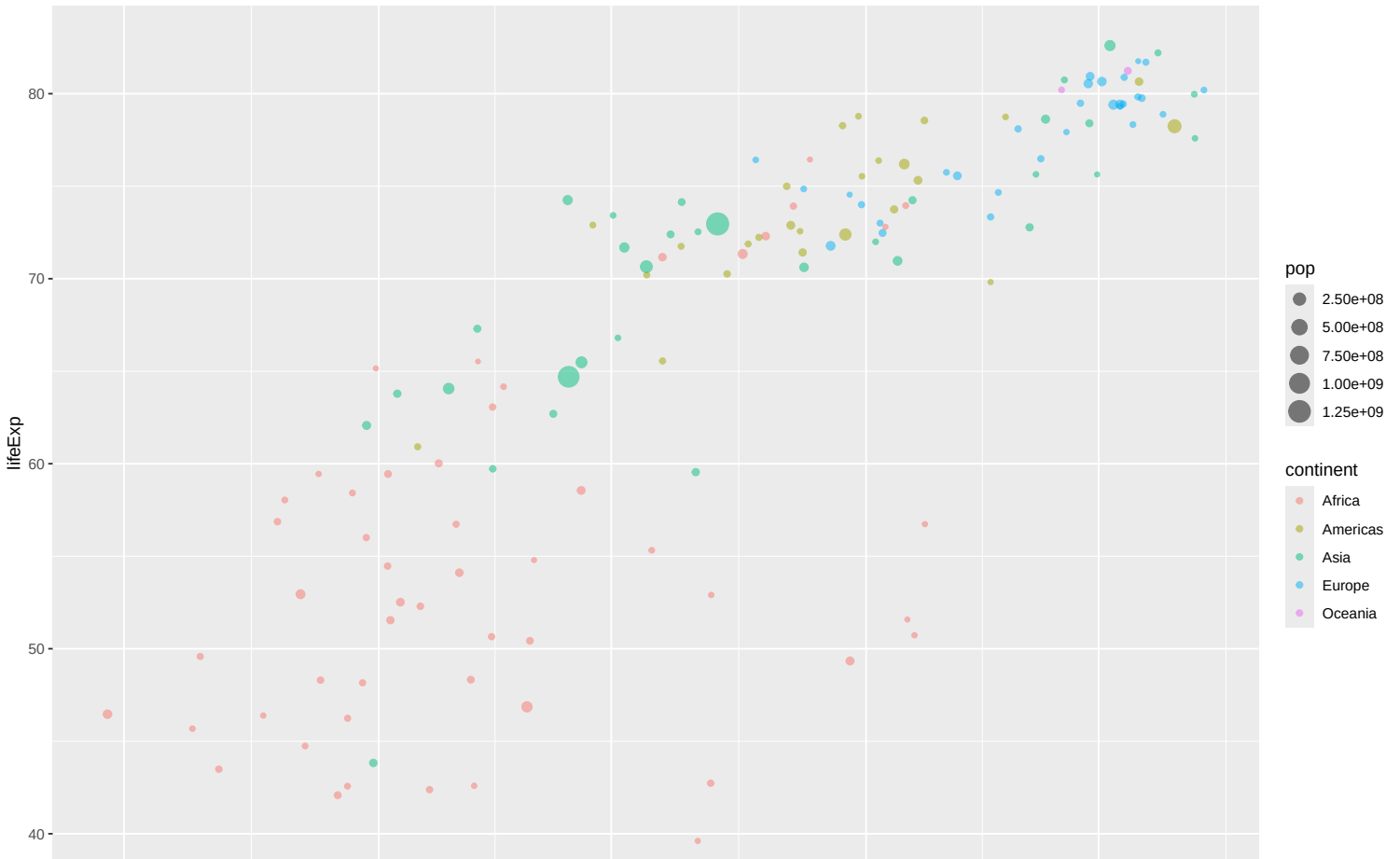
```
## Warning: package 'ggrepel' was built under R version 4.4.3

## Warning: ggrepel: 6 unlabeled data points (too many overlaps). C
## increasing max.overlaps
```



Scatter Plot: Log X

```
ggplot(gapminder_2007,
       aes(x=gdpPercap,y = lifeExp,size=pop,color=continent)) +
geom_point(alpha=0.5)+scale_x_log10()
```



Scatter Plot: by group Facet

```
ggplot(gapminder_2007,
       aes(x=gdpPercap, y = lifeExp,size = pop,
           color=continent)) +
geom_point(alpha=0.5) +
scale_x_log10()+
facet_wrap(~ continent)
```

Scatter Plot: by group Facet

Scatter Plot: by group Facet

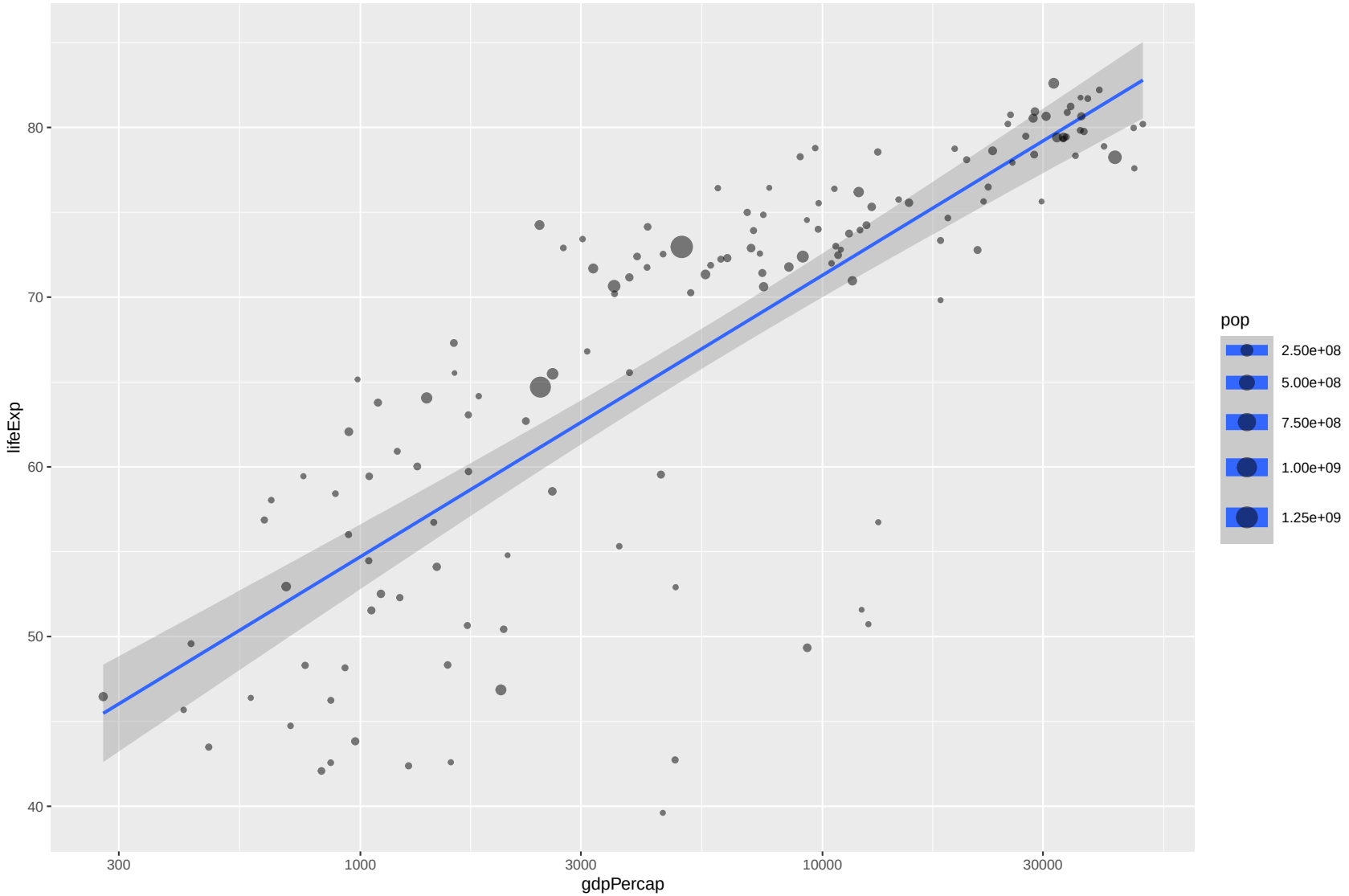
```
ggplot(gapminder,aes(x=gdpPercap, y = lifeExp,size = pop,
                     color = continent)) +
geom_point() +scale_x_log10()+facet_wrap(~ year)
```

Scatter Plot: by group Facet

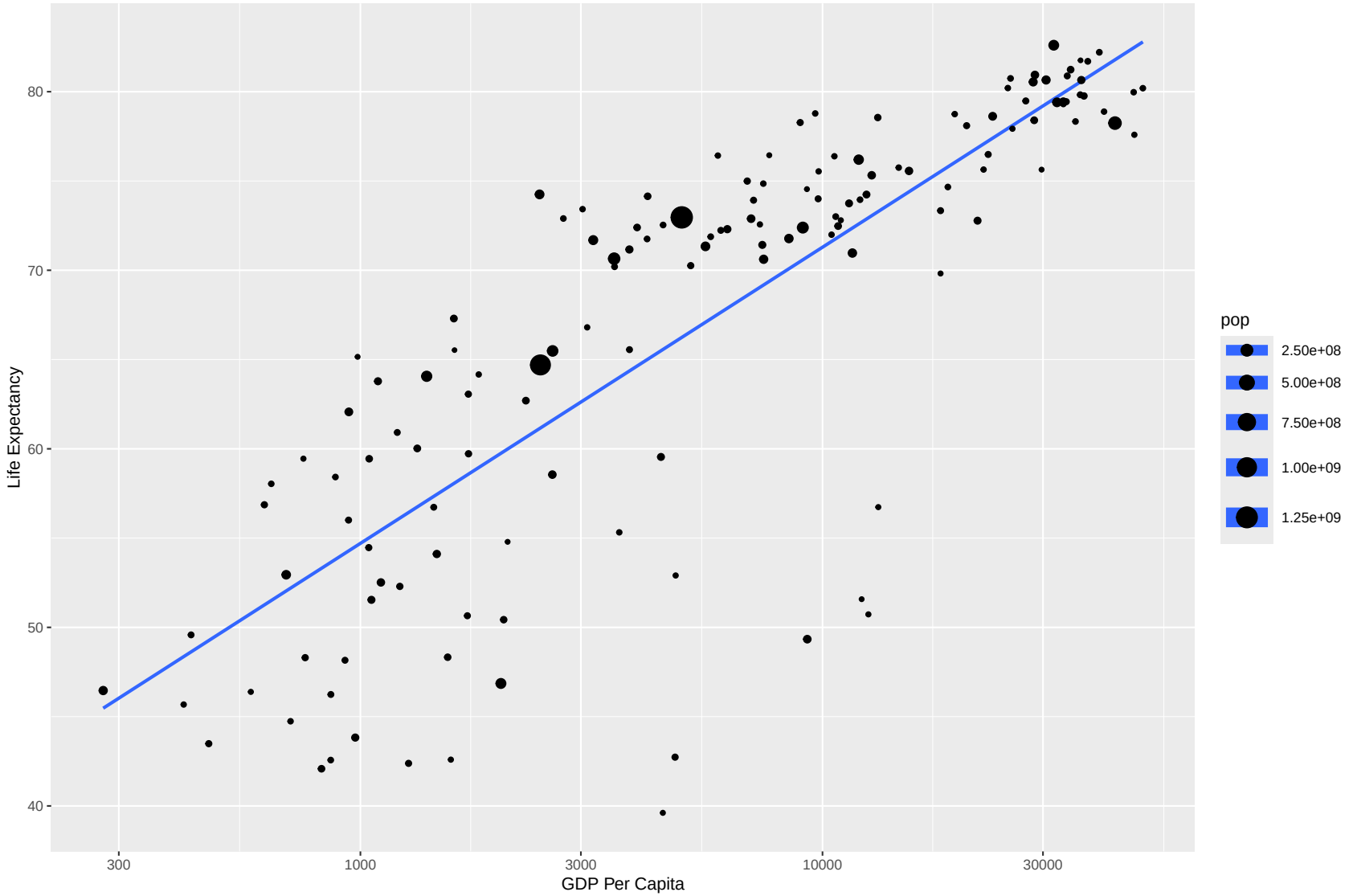
Scatter Plot: regression line

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(x=gdpPercap, y = lifeExp,size = pop)) +
    geom_smooth(method = lm) +
    geom_point(alpha=0.5) +
    scale_x_log10()
```

Scatter Plot: regression line

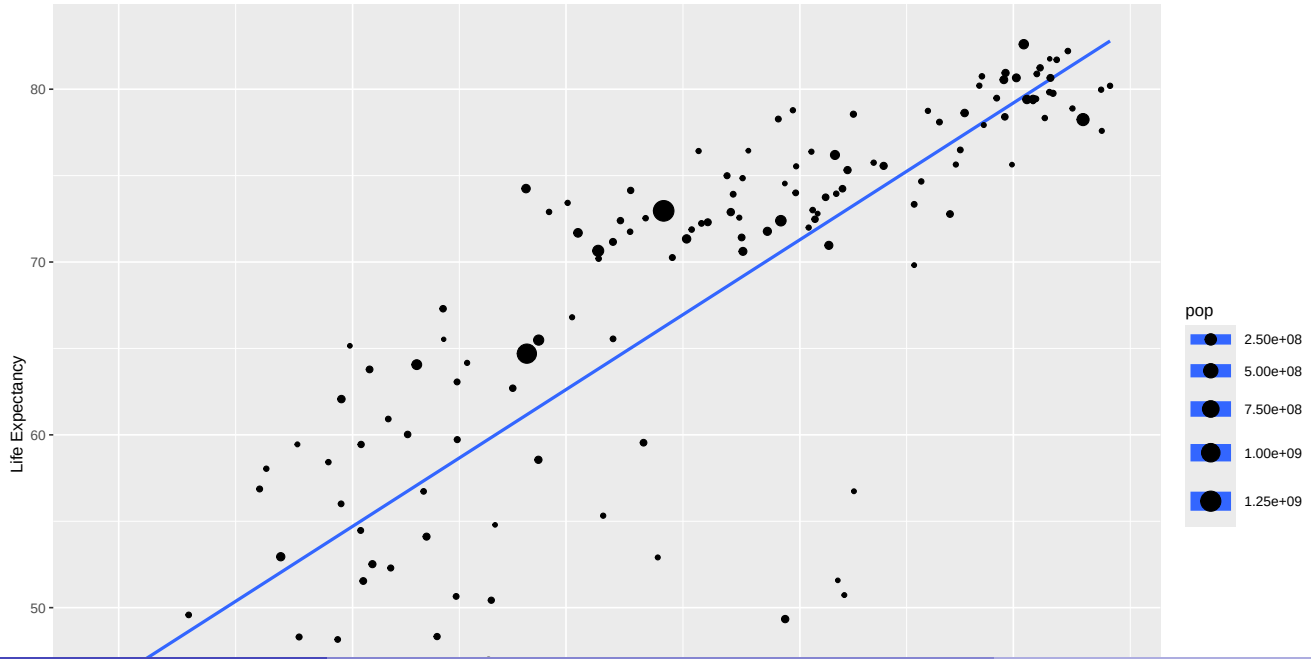


Scatter Plot: Straight line



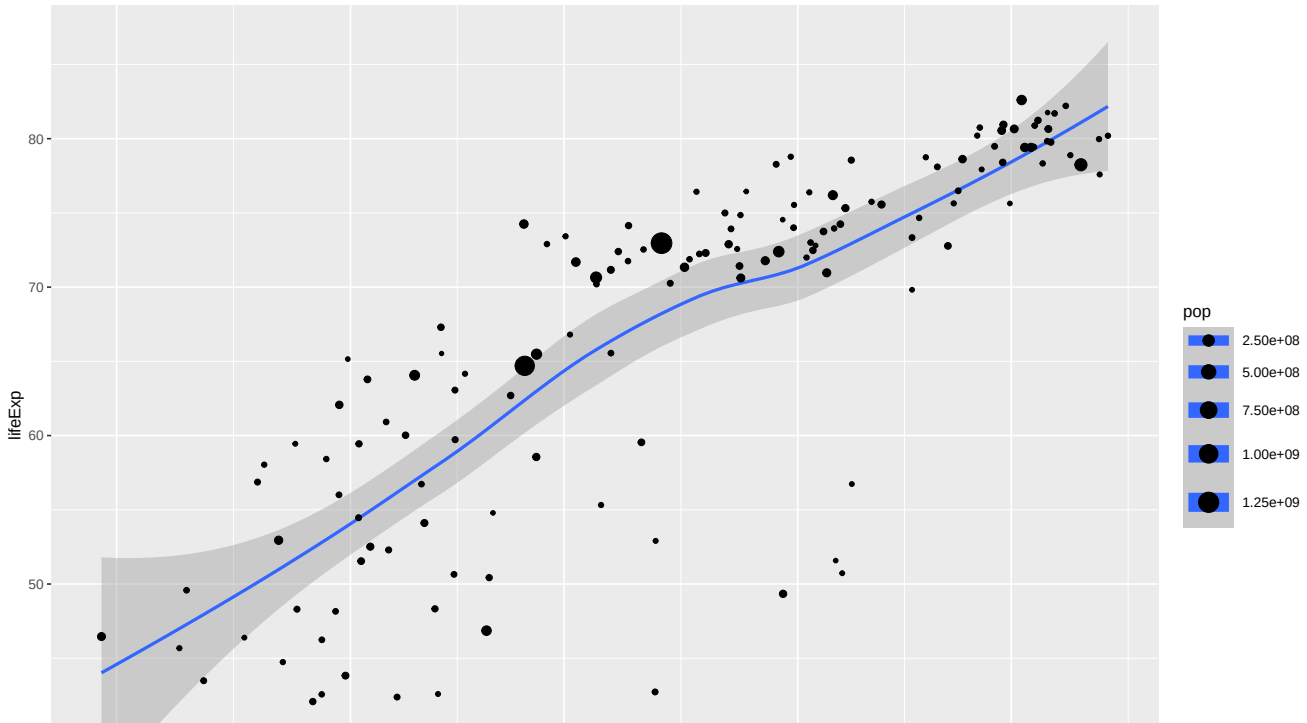
Scatter Plot:

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(x=gdpPercap, y = lifeExp,size = pop)) +
    geom_smooth(method = lm,se=FALSE) +
    geom_point() +
    scale_x_log10()+
    labs(x="GDP Per Capita",y="Life Expectancy")
```



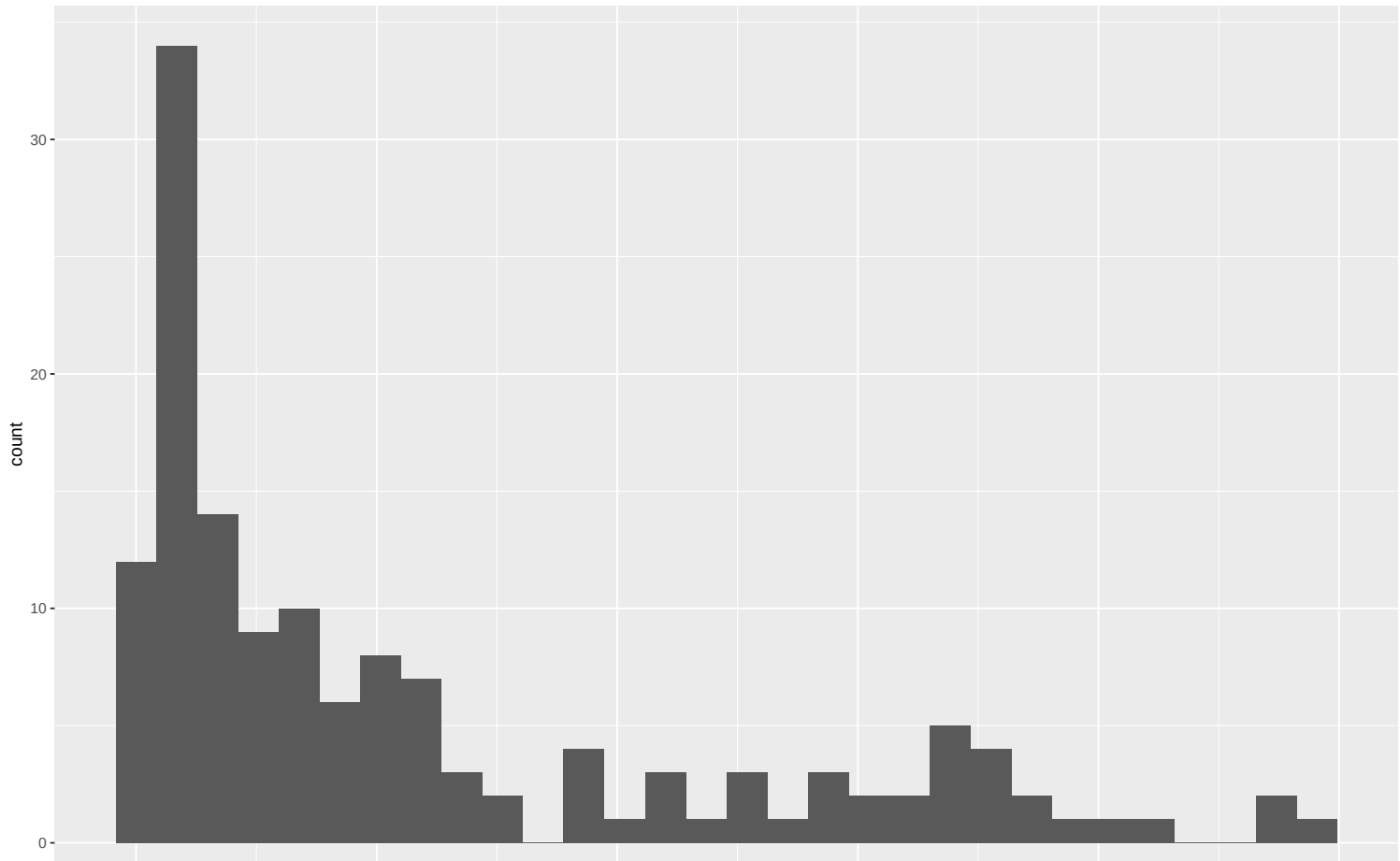
Scatter Plot: smoothing line

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(x=gdpPercap, y = lifeExp,size = pop)) +
    geom_smooth() +
    geom_point() +
    scale_x_log10()
```



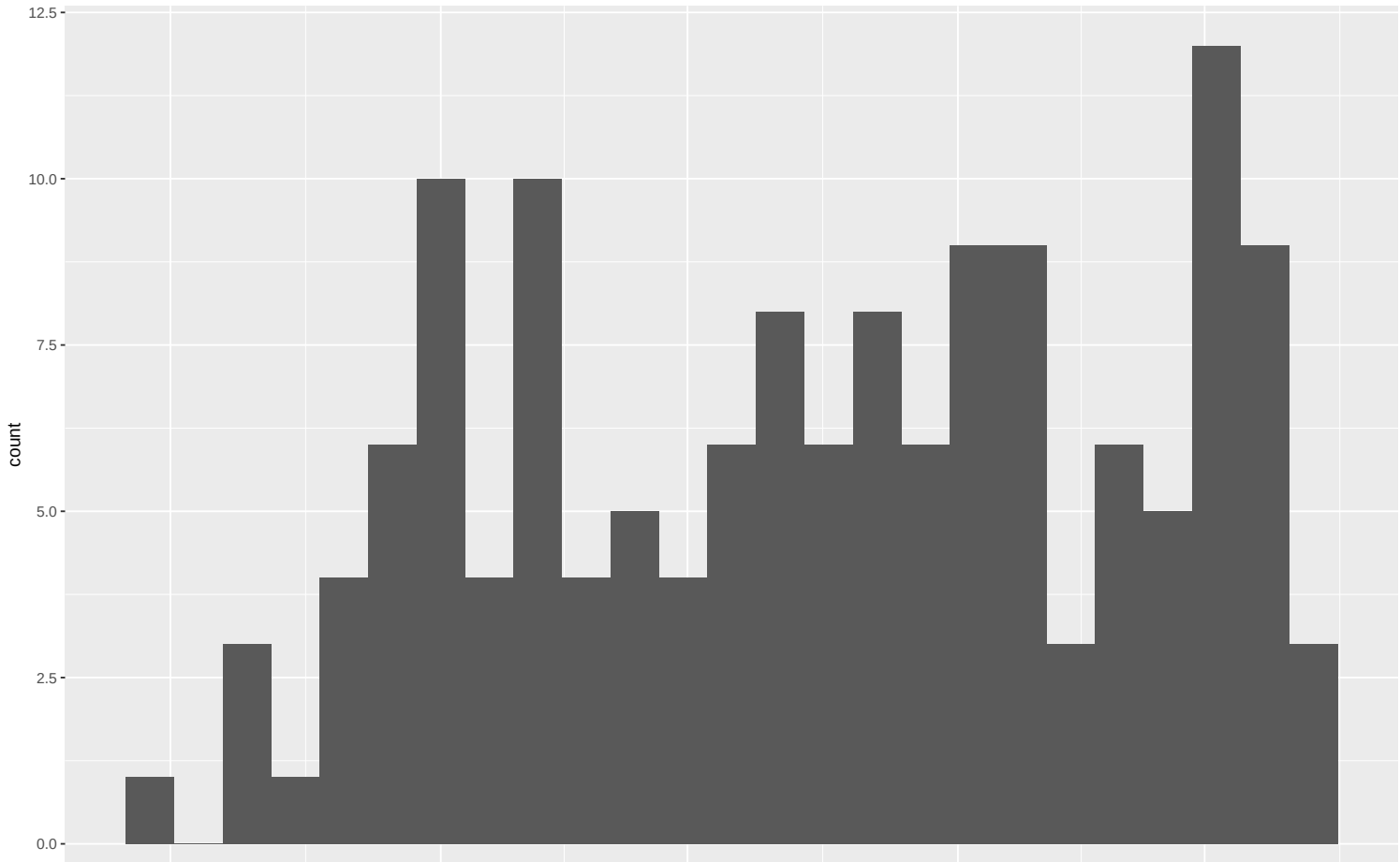
Histogram

```
gapminder %>%
    filter(year==2007) %>% ggplot(aes(x=gdpPercap))+
    geom_histogram()
```



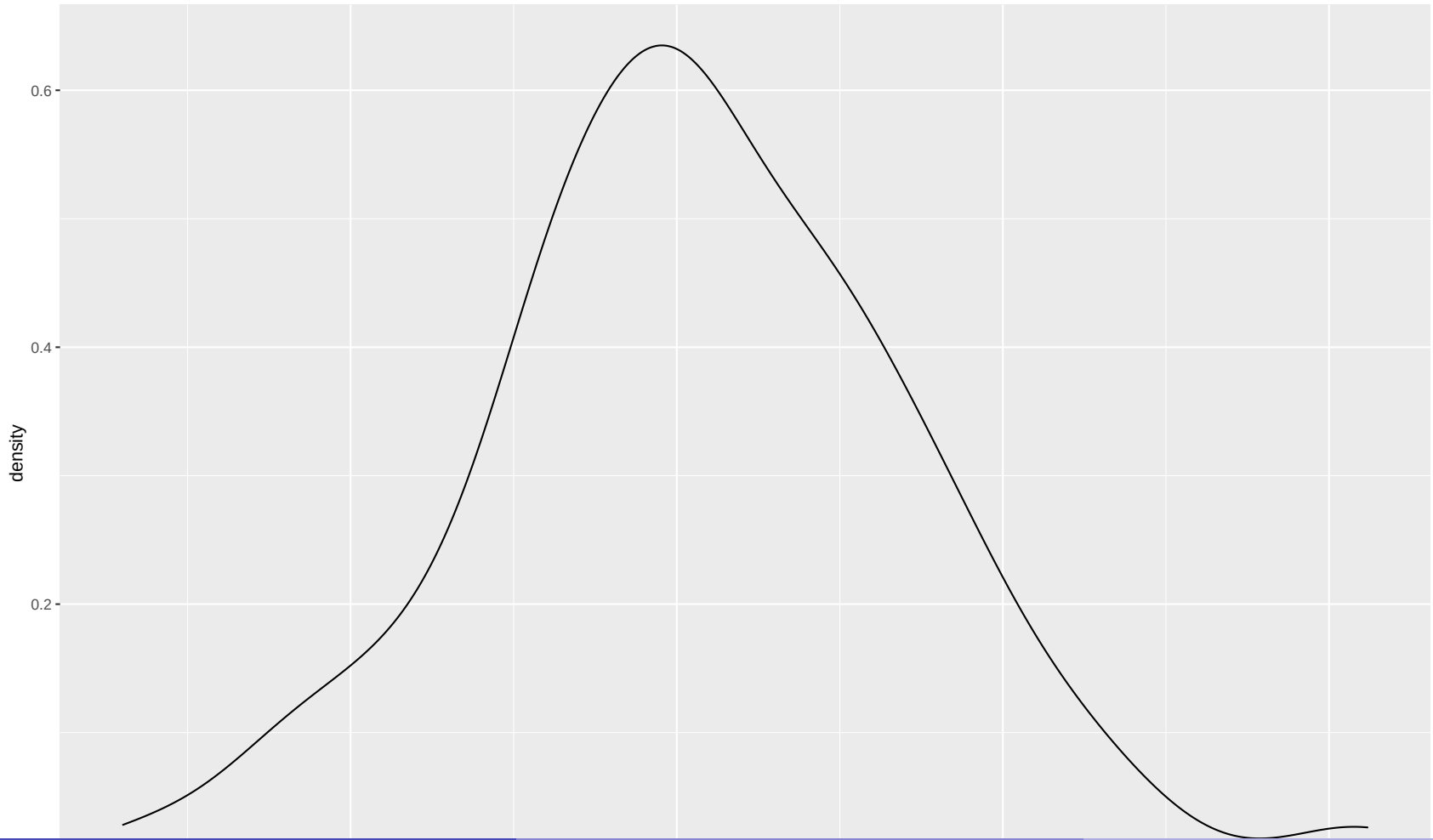
Histogram

```
gapminder %>%
    filter(year==2007) %>% ggplot(aes(x=gdpPercap))+
    geom_histogram(bins=25) + scale_x_log10()
```



Kdensity

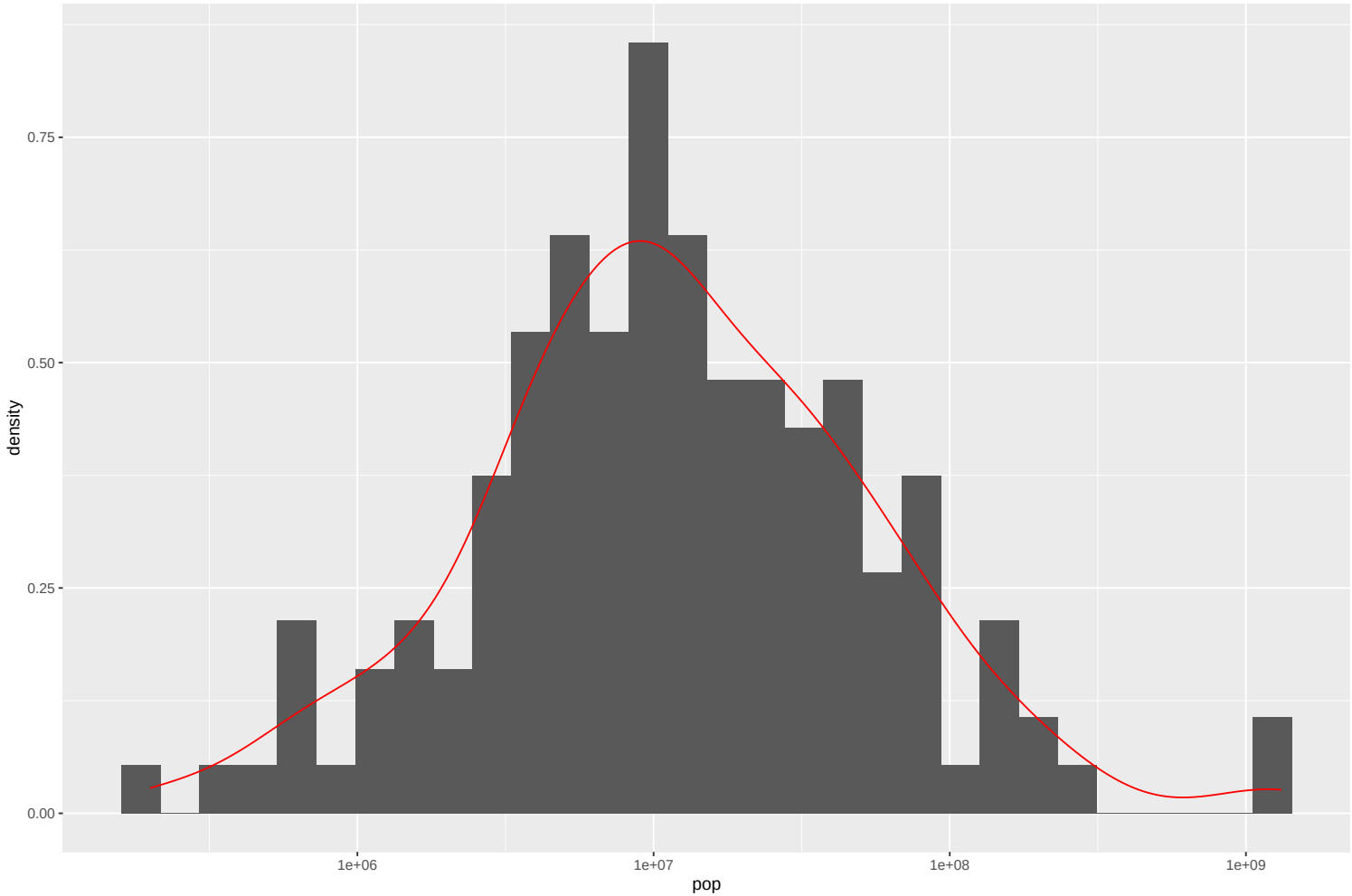
```
gapminder %>% filter(year==2007) %>% ggplot(aes(x=pop)) +
    geom_density() + scale_x_log10()
```



Kdensity+Histogram

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(x=pop)) +
    geom_histogram(aes(y = stat(density)))+
    geom_density(col="red") +
    scale_x_log10()
```

Kdensity+Histogram

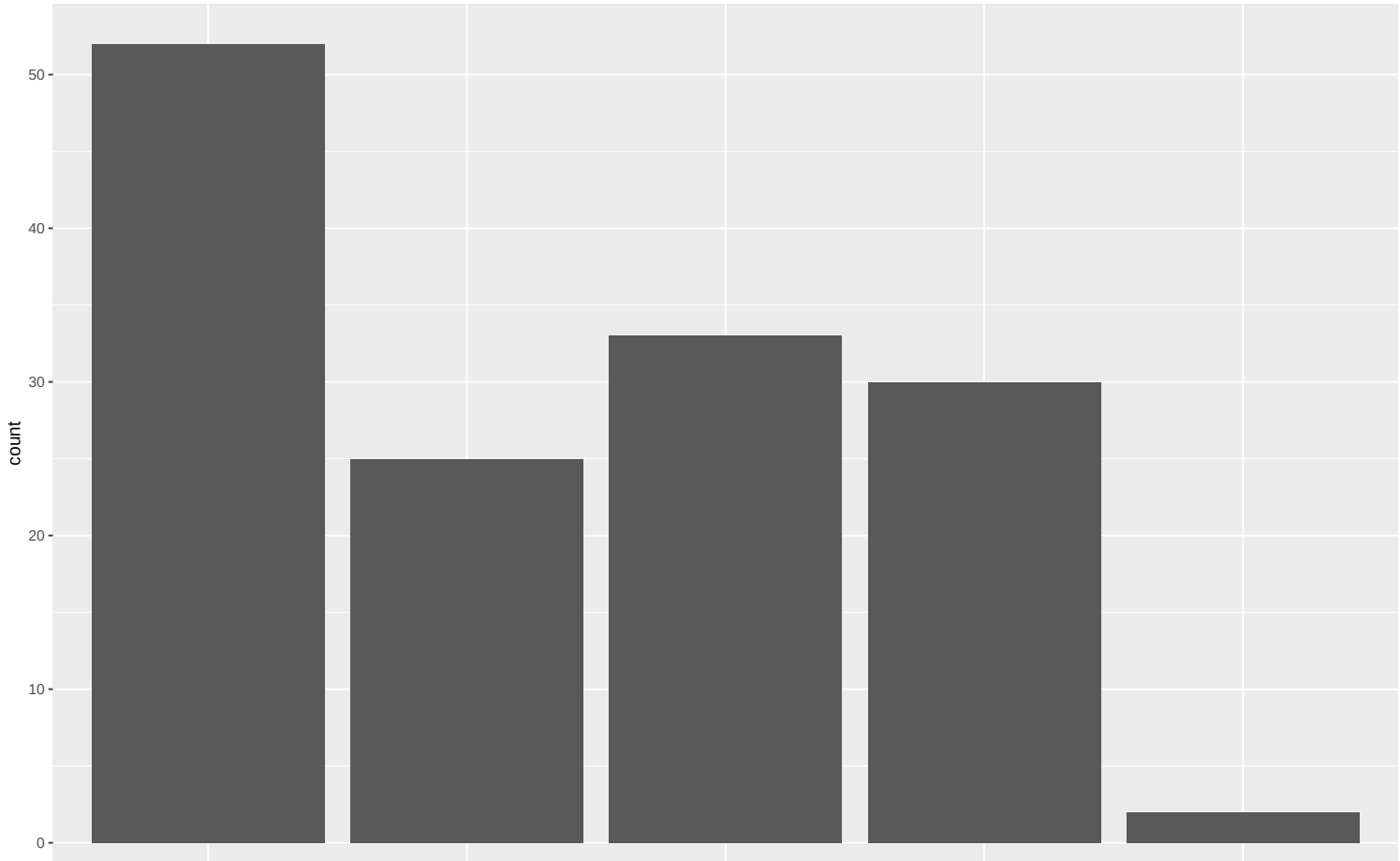


Bar

- `geom_bar()` and `geom_col()`.
- `geom_bar()` makes the height of the bar proportional to the number of cases in each group. It uses `stat_count()` by default: it counts the number of cases at each x position.
- If you want the heights of the bars to represent values in the data, use `geom_col()` instead. It uses stat_identity(): it leaves the data as is.

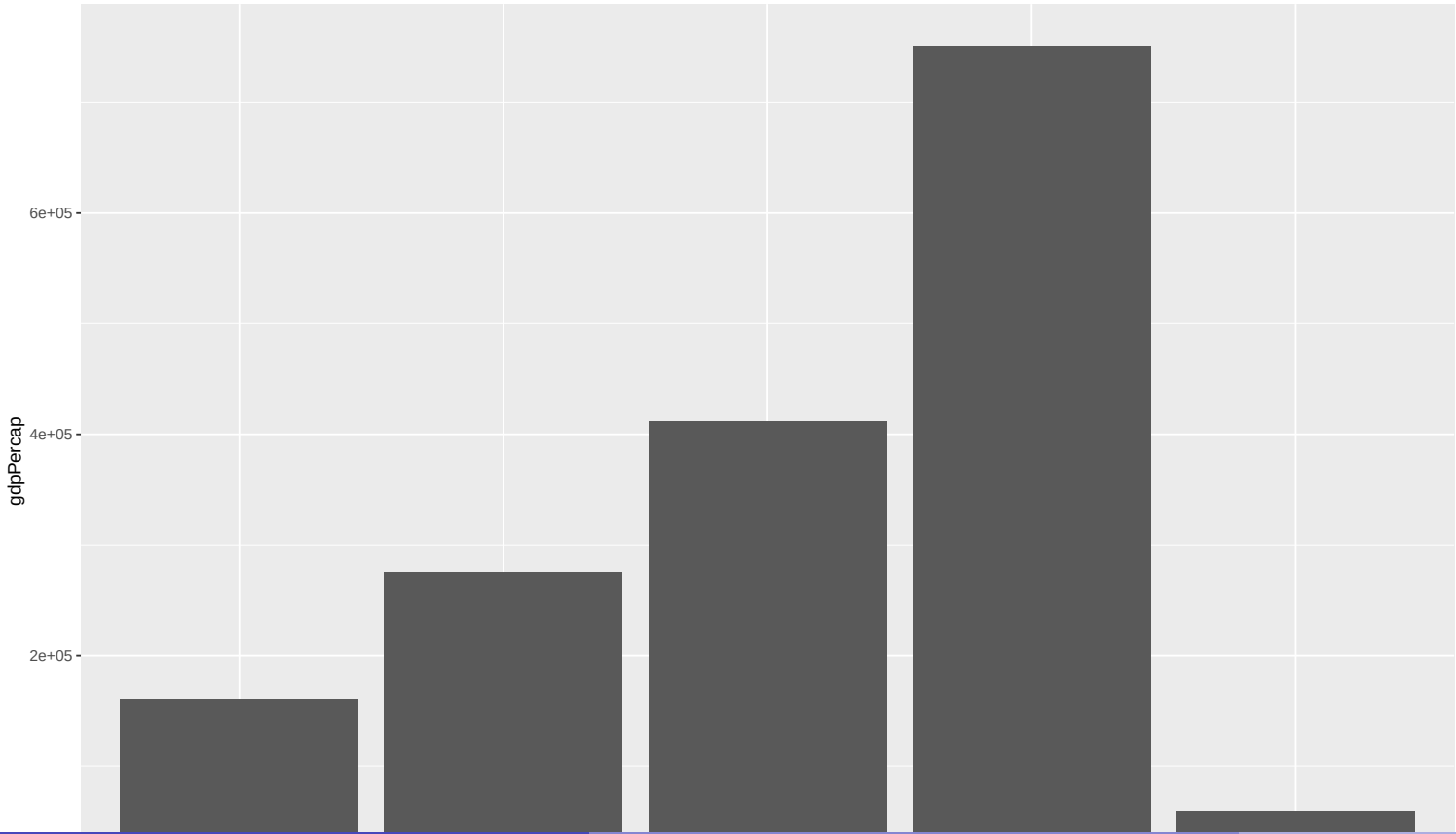
Bar

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(x=continent)) + geom_bar()
```



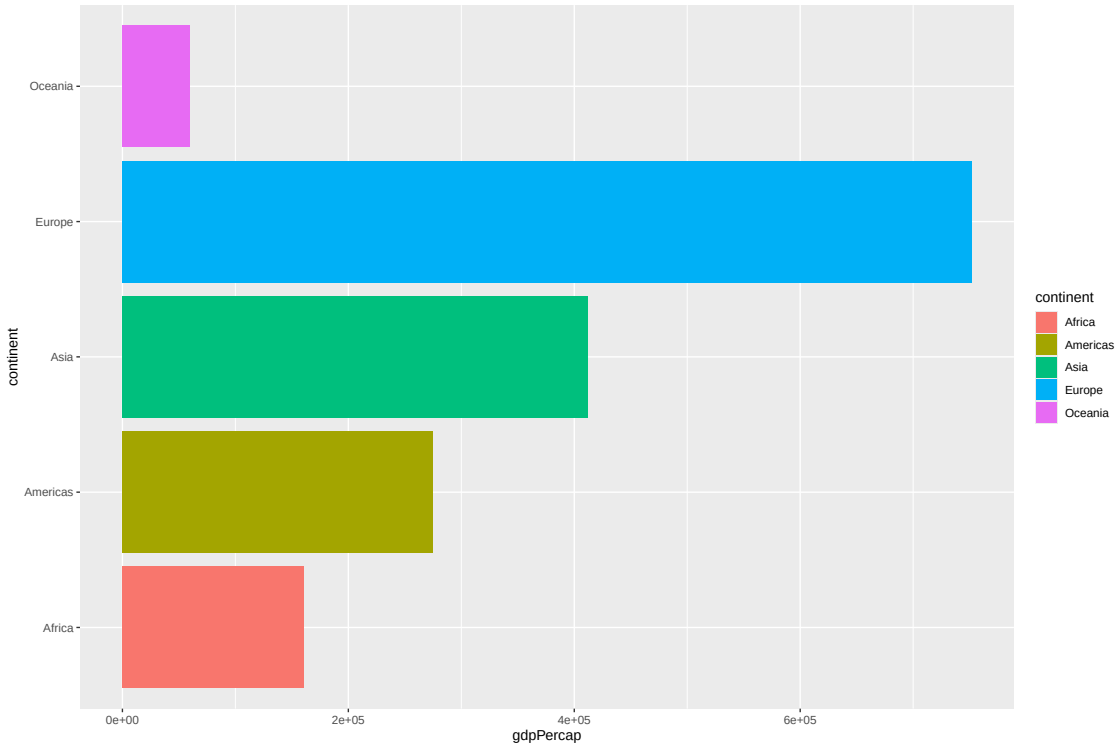
Bar

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(y=gdpPercap,x=continent)) +
    geom_col()
```



Bar

```
gapminder %>%
    filter(year==2007) %>%
    ggplot(aes(y=gdpPercap,x=continent,fill=continent)) +
    geom_col() + coord_flip()
```



Reference

1. Grolemund & Wickham(2017), R for Data Science
2. DataCamp(2018), Introduction to Tidyverse
3. John Sullivan(2019), Data Cleaning with R and the Tidyverse: Detecting Missing Values