# Block 4: Webbanalys - facit

*Simon Löfwander*

*2020-01-08*

## Del 1 - Analysera Google Analytics data

I denna övning ska du svara på ett antal frågor kopplade till data som återfinns i Google Analytics. Det finns ett flertal olika data set att välja mellan när du ska besvara varje fråga, där varje data set tillhör en eller flera frågor.

## 1. Vem har besökt sidan?

Denna sektion innehåller frågor som är kopplade till vem som har besökt sidan - alltså information om vilka besökarna *är*.

Allra först börjar vi med att importera de bibliotek som kommer att användas för att kunna utföra övningarna.

```
if(!"pacman" %in% installed.packages()[,"Package"]) install.packages("pacman")
pacman::p_load(tidyverse, lubridate, CausalImpact, scales)
```

### 1.a. Vilket land hade flest sessioner perioden 2019-06-01 till 2019-08-31?

```
read_csv("country_data.csv") -> location_tbl

location_tbl %>% glimpse()
```

```
## Observations: 2,156
## Variables: 11
## $ country                   <chr> "United States", "United States", "Uni...
## $ month_of_year             <date> 2019-12-31, 2019-05-31, 2019-01-31, 2...
## $ users                     <dbl> 25774, 24922, 23874, 23718, 23703, 235...
## $ new_users                 <dbl> 22714, 21394, 19882, 20974, 20794, 203...
## $ sessions                  <dbl> 35348, 35392, 33424, 32888, 32048, 326...
## $ bounce_rate               <dbl> 0.3172740, 0.2934561, 0.2955361, 0.315...
## $ pages_per_session         <dbl> 5.943646, 5.582872, 5.750628, 5.829816...
## $ avg__session_duration     <dbl> 225.66793, 214.31583, 219.29239, 224.5...
## $ transactions              <dbl> 146, 80, 69, 103, 94, 91, 82, 53, 95, ...
## $ revenue                   <dbl> 7311.70, 4565.06, 3766.71, 4212.42, 40...
## $ ecommerce_conversion_rate <dbl> 0.0041303610, 0.0022603978, 0.00206438...
```

```
location_tbl %>%
  filter(month_of_year >= as.Date("2019-06-01") &
           month_of_year <= as.Date("2019-09-01")) %>%
  group_by(country) %>%
  summarise(sessions = sum(sessions)) %>%
```

```r
  arrange(desc(sessions)) %>%
  slice(1) %>%
  pull(country) %>%
  print()
```

```
## [1] "United States"
```

### 1.b. Vilket land hade högst konverteringsgrad under 2019? *(ecommerce conversion rate = transactions / sessions)*

```r
location_tbl %>%
  mutate(year = lubridate::year(month_of_year)) %>%
  filter(year %in% 2019) %>%
  group_by(country) %>%
  summarise(sessions = sum(sessions),
            transactions = sum(transactions)) %>%
  filter(transactions > 0) %>%
  mutate(ecommerce_conversion_rate = transactions / sessions) %>%
  arrange(desc(ecommerce_conversion_rate)) %>%
  head() %>%
  print()
```

```
## # A tibble: 6 x 4
##   country              sessions transactions ecommerce_conversion_rate
##   <chr>                   <dbl>        <dbl>                     <dbl>
## 1 Paraguay                  139            2                   0.0144
## 2 Puerto Rico               350            3                   0.00857
## 3 Kuwait                    288            2                   0.00694
## 4 Cambodia                  235            1                   0.00426
## 5 United States          392234         1078                   0.00275
## 6 United Arab Emirates     1788            2                   0.00112
```

### 1.c. Går det att dra några slutsatser av resultatet i *b)* - i så fall vad?

### 1.d. Vilken åldersgrupp hade högst genomsnittliga transaktionsvärde under 2019?

```r
read_csv("age_bracket.csv") -> age_tbl

age_tbl %>% glimpse()
```

```
## Observations: 2,062
## Variables: 11
## $ age                       <chr> "25-34", "25-34", "25-34", "25-34", "2...
## $ date                      <date> 2019-12-02, 2019-08-07, 2019-08-20, 2...
## $ users                     <dbl> 845, 803, 745, 743, 732, 722, 720, 713...
## $ new_users                 <dbl> 709, 642, 620, 597, 574, 568, 584, 570...
```

```
## $ sessions                  <dbl> 969, 901, 847, 818, 817, 831, 797, 843...
## $ bounce_rate               <dbl> 0.4138287, 0.3485017, 0.4935065, 0.334...
## $ pages_per_session         <dbl> 4.023736, 4.532741, 4.121606, 4.881418...
## $ avg__session_duration     <dbl> 140.6842, 179.2542, 156.1806, 194.8191...
## $ transactions              <dbl> 1, 0, 0, 0, 0, 2, 1, 0, 0, 0, 0, 0, 0,...
## $ revenue                   <dbl> 41.80, 0.00, 0.00, 0.00, 0.00, 55.97, ...
## $ ecommerce_conversion_rate <dbl> 0.001031992, 0.000000000, 0.000000000,...
```

```r
age_tbl %>%
  group_by(age) %>%
  summarise(transactions = sum(transactions),
            revenue = sum(revenue)) %>%
  mutate(revenue_per_transaction = revenue / transactions) %>%
  arrange(desc(revenue_per_transaction)) %>%
  print()
```

```
## # A tibble: 6 x 4
##   age   transactions revenue revenue_per_transaction
##   <chr>        <dbl>   <dbl>                   <dbl>
## 1 55-64           14   1377.                    98.4
## 2 45-54           20   1281.                    64.0
## 3 35-44           33   2068.                    62.7
## 4 25-34           55   2548.                    46.3
## 5 65+              3     95.0                   31.7
## 6 18-24           26    734.                    28.2
```

### 1.e. Vilken åldersgrupp hade högst konverteringsgrad under 2019?

```r
 age_tbl %>%
  group_by(age) %>%
  summarise(transactions = sum(transactions),
            sessions = sum(sessions)) %>%
  mutate(e_commerce_conversion_rate = transactions / sessions) %>%
  arrange(desc(e_commerce_conversion_rate))
```

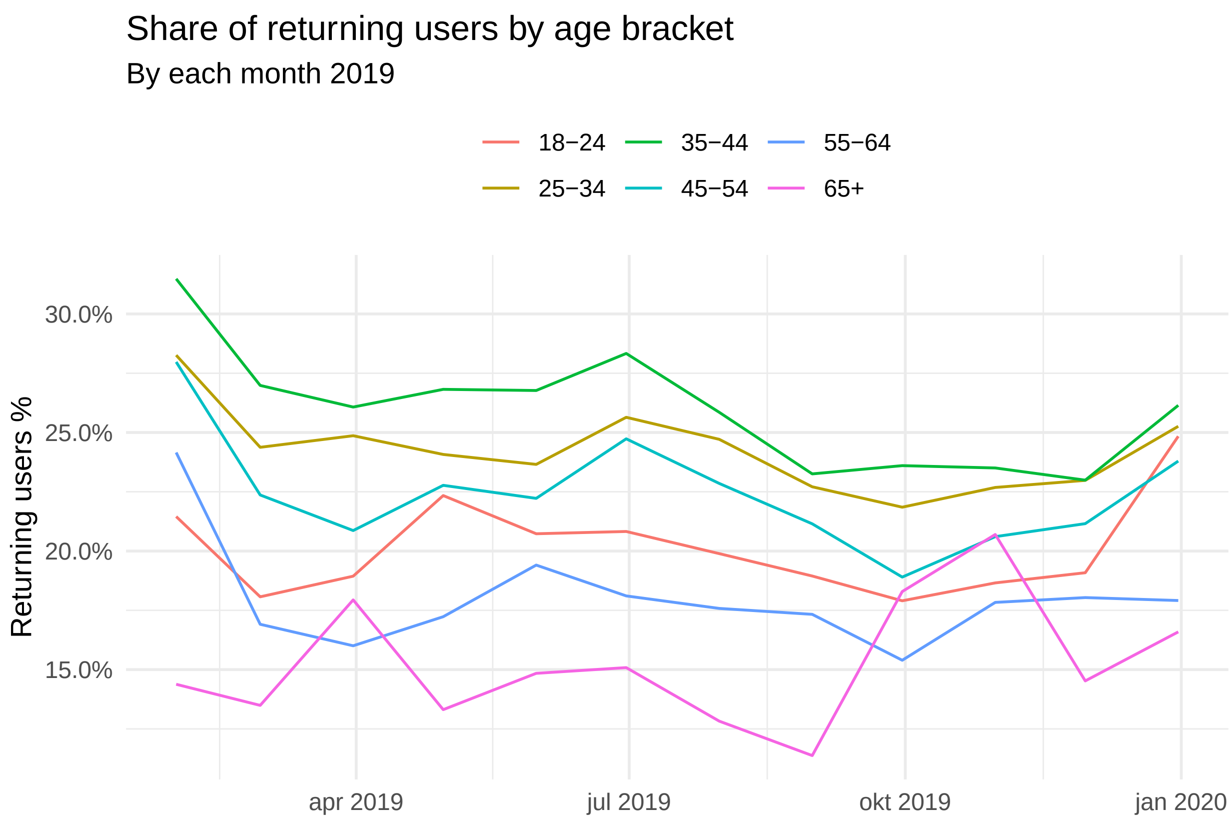
```
## # A tibble: 6 x 4
##   age   transactions sessions e_commerce_conversion_rate
##   <chr>        <dbl>    <dbl>                      <dbl>
## 1 55-64           14     9324                   0.00150
## 2 45-54           20    26342                   0.000759
## 3 65+              3     4549                   0.000659
## 4 35-44           33    76682                   0.000430
## 5 18-24           26    72369                   0.000359
## 6 25-34           55   174465                   0.000315
```

### 1.d. Visualisera andelen återvändande besökare per månad och åldersgrupp under 2019. *(returning user = user - new user)*

```r
age_tbl %>%
  group_by(age, month = as.Date(cut(date, "month"))) %>%
  summarise(users = sum(users),
            new_users = sum(new_users)) %>%
  ungroup() %>%
  mutate(year = lubridate::year(month)) %>%
  filter(year %in% 2019) %>%
  mutate(month = month %>%
           ceiling_date(., "month") - days(1),
         returning_users = (users - new_users)/users %>%
           round(.,2)) -> returning_tbl

returning_tbl %>% glimpse()
```

```
## Observations: 72
## Variables: 6
## $ age             <chr> "18-24", "18-24", "18-24", "18-24", "18-24", "18...
## $ month           <date> 2019-01-31, 2019-02-28, 2019-03-31, 2019-04-30,...
## $ users           <dbl> 4451, 4571, 5818, 5801, 5562, 5181, 5369, 5737, ...
## $ new_users       <dbl> 3496, 3745, 4716, 4505, 4409, 4102, 4301, 4650, ...
## $ year            <dbl> 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, ...
## $ returning_users <dbl> 0.2145585, 0.1807044, 0.1894122, 0.2234098, 0.20...
```

```r
returning_tbl %>%
  ggplot(aes(x = month, y = returning_users, color = age)) +
  geom_line() +
  scale_y_continuous(labels = scales::percent) +
  labs(
    x = "",
    y = "Returning users %",
    title = "Share of returning users by age bracket",
    subtitle = "By each month 2019",
    color = ""
  ) +
  theme_minimal() +
  theme(legend.position = "top")
```




## 2. Hur kom de till sidan?

### 2.a. Från vilken trafikkälla kom flest transaktioner under 2019?

```
read_csv("traffic_source.csv") -> traffic_source_tbl

traffic_source_tbl %>% glimpse()
```

```
## Observations: 104
## Variables: 11
## $ default_channel_grouping  <chr> "Organic Search", "Organic Search", "O...
## $ month_of_year             <date> 2019-11-30, 2019-09-30, 2019-05-31, 2...
## $ users                     <dbl> 35124, 34508, 33554, 32464, 32068, 315...
## $ new_users                 <dbl> 31778, 31618, 30015, 29388, 28752, 284...
## $ sessions                  <dbl> 42264, 41573, 40375, 39058, 38272, 380...
## $ bounce_rate               <dbl> 0.5371711, 0.5233445, 0.5020186, 0.504...
## $ pages_per_session         <dbl> 3.585652, 3.666226, 3.853152, 3.975984...
## $ avg__session_duration     <dbl> 144.1928, 143.5604, 146.0001, 153.3551...
## $ ecommerce_conversion_rate <dbl> 0.0014433087, 0.0015154066, 0.00131269...
## $ transactions              <dbl> 61, 63, 53, 77, 57, 53, 90, 71, 69, 59...
## $ revenue                   <dbl> 2726.90, 2872.67, 3156.82, 3816.18, 33...
```

```r
traffic_source_tbl %>%
  mutate(year = lubridate::year(month_of_year)) %>%
  filter(year %in% 2019) %>%
  group_by(default_channel_grouping) %>%
  summarise(transactions = sum(transactions)) %>%
  arrange(desc(transactions)) %>%
  print()
```

```
## # A tibble: 8 x 2
##   default_channel_grouping transactions
##   <chr>                           <dbl>
## 1 Organic Search                    739
## 2 Direct                            255
## 3 Paid Search                       107
## 4 (Other)                            38
## 5 Social                              6
## 6 Referral                            4
## 7 Affiliates                          2
## 8 Display                             0
```

### 2.b. Vilken trafikkälla hade mest engagerade besökare i snitt under December månad? *(engagement = pages / session)*

```r
traffic_source_tbl %>%
  mutate(year = lubridate::year(month_of_year)) %>%
  filter(month_of_year == as.Date("2019-12-31")) %>%
  dplyr::select(default_channel_grouping, pages_per_session) %>%
  arrange(desc(pages_per_session)) %>%
  head() %>%
  print()
```

```
## # A tibble: 6 x 2
##   default_channel_grouping pages_per_session
##   <chr>                                <dbl>
## 1 Referral                              7.14
## 2 Paid Search                           6.24
## 3 (Other)                               4.79
## 4 Direct                                4.71
## 5 Organic Search                        3.70
## 6 Social                                3.11
```

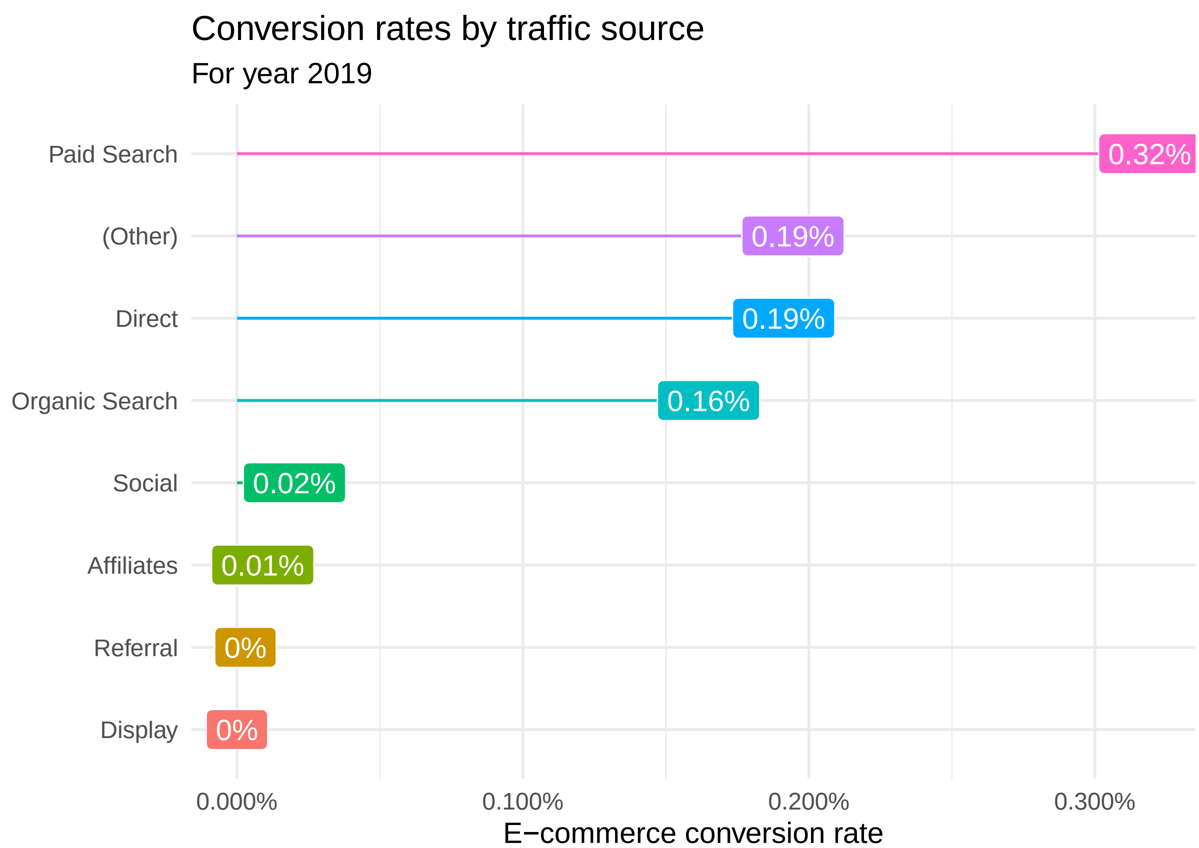
### 2.c. Visualisera genomsnittlig konverteringsgrad per trafikkälla för 2019.

```r
traffic_source_tbl %>% glimpse()
```

```
## Observations: 104
## Variables: 11
## $ default_channel_grouping  <chr> "Organic Search", "Organic Search", "O...
```

```
## $ month_of_year             <date> 2019-11-30, 2019-09-30, 2019-05-31, 2...
## $ users                     <dbl> 35124, 34508, 33554, 32464, 32068, 315...
## $ new_users                 <dbl> 31778, 31618, 30015, 29388, 28752, 284...
## $ sessions                  <dbl> 42264, 41573, 40375, 39058, 38272, 380...
## $ bounce_rate               <dbl> 0.5371711, 0.5233445, 0.5020186, 0.504...
## $ pages_per_session         <dbl> 3.585652, 3.666226, 3.853152, 3.975984...
## $ avg__session_duration     <dbl> 144.1928, 143.5604, 146.0001, 153.3551...
## $ ecommerce_conversion_rate <dbl> 0.0014433087, 0.0015154066, 0.00131269...
## $ transactions              <dbl> 61, 63, 53, 77, 57, 53, 90, 71, 69, 59...
## $ revenue                   <dbl> 2726.90, 2872.67, 3156.82, 3816.18, 33...
```

```
traffic_source_tbl %>%
  mutate(year = lubridate::year(month_of_year)) %>%
  filter(year %in% 2019) %>%
  group_by(default_channel_grouping) %>%
  summarise(transactions = sum(transactions),
            sessions = sum(sessions)) %>%
  mutate(conversion_rate = transactions/sessions,
         default_channel_grouping = fct_reorder(default_channel_grouping,
                                               conversion_rate)) %>%
  ggplot(aes(x = default_channel_grouping,
             y = conversion_rate,
             color = default_channel_grouping,
             fill = default_channel_grouping)) +
  geom_linerange(aes(x = default_channel_grouping,
                     ymin = 0,
                     ymax = conversion_rate)) +
  geom_label(aes(label = paste0(round(conversion_rate*100,2),"%") ),
             color = "white") +
  scale_y_continuous(labels = scales::percent) +
  coord_flip() +
  labs(
    x = "",
    y = "E-commerce conversion rate",
    title = "Conversion rates by traffic source",
    subtitle = "For year 2019"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```




## 3. Vad gjorde de på sidan?

### 3.a. Vilken landningssida med minst 1000 sessions hade högst bounce rate under december månad 2019?

```
read_csv("landing_page_tbl.csv") -> landing_page_tbl

landing_page_tbl %>% glimpse()
```

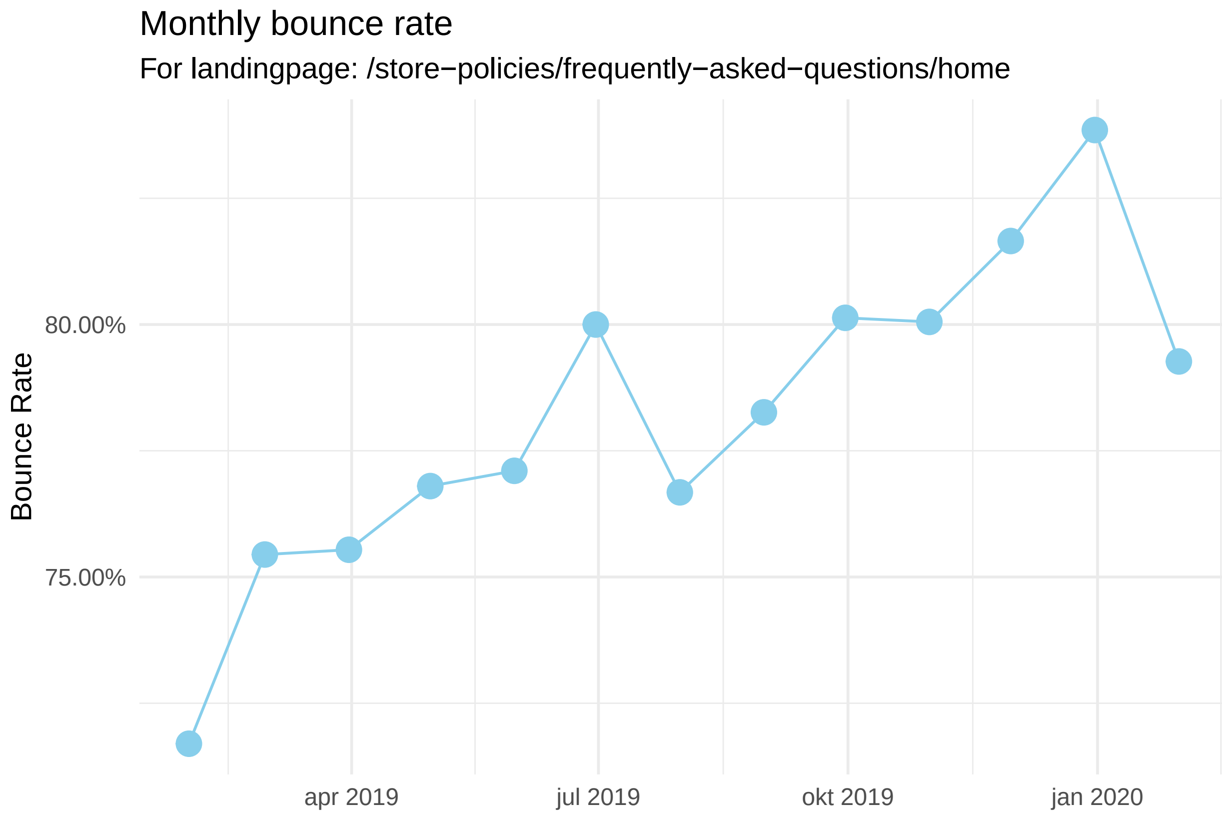
```
## Observations: 5,001
## Variables: 11
## $ landing_page              <chr> "/home", "/home", "/home", "/home", "/...
## $ month_of_year             <date> 2019-11-30, 2019-10-31, 2019-09-30, 2...
## $ sessions                  <dbl> 43346, 39561, 38849, 38145, 38034, 367...
## $ `%_new_sessions`          <dbl> 0.7355004, 0.7347388, 0.7499807, 0.716...
## $ new_users                 <dbl> 31881, 29067, 29136, 27332, 27208, 255...
## $ bounce_rate               <dbl> 0.4587736, 0.4471576, 0.4533965, 0.414...
## $ pages_per_session         <dbl> 4.264430, 3.830894, 4.348580, 4.695347...
## $ avg__session_duration     <dbl> 176.69294, 168.22919, 170.07913, 182.0...
## $ transactions              <dbl> 21, 20, 23, 47, 28, 35, 25, 39, 31, 22...
## $ revenue                   <dbl> 1286.75, 782.10, 1710.78, 2624.50, 268...
## $ ecommerce_conversion_rate <dbl> 0.0004844738, 0.0005055484, 0.00059203...
```

```r
landing_page_tbl %>%
  filter(month_of_year %in% as.Date("2019-12-31")) %>%
  filter(sessions >= 1000) %>%
  arrange(desc(bounce_rate)) %>%
  slice(1) %>%
  pull(landing_page) -> highest_bounce_rate_page

highest_bounce_rate_page
```

```
## [1] "/store-policies/frequently-asked-questions/home"
```

### 3.b. Visualisera bounce rate varje månad för den landningssida du angav i 3.a.

```r
landing_page_tbl %>%
  filter(landing_page %in% highest_bounce_rate_page) %>%
  ggplot(aes(x = month_of_year, y = bounce_rate)) +
  geom_point(size = 4, color = "skyblue") +
  geom_line(color = "skyblue") +
  scale_y_continuous(labels = scales::percent) +
  labs(
    x = "",
    y = "Bounce Rate",
    title = "Monthly bounce rate",
    subtitle =  paste("For landingpage:", highest_bounce_rate_page)
  ) +
  theme_minimal()
```




## 4. Konverterade de?

### 4.a. Under vilken månad såldes flest produkter under 2019?

```
read_csv("products_tbl.csv") -> products_tbl

products_tbl %>% glimpse()
```

```
## Observations: 2,915
## Variables: 11
## $ product               <chr> "Google Color Block Notebook", "Google Cam...
## $ month_of_year         <date> 2019-10-31, 2019-10-31, 2019-07-31, 2019-...
## $ product_revenue       <dbl> 1812.00, 1661.00, 935.61, 793.00, 599.95, ...
## $ unique_purchases      <dbl> 2, 2, 2, 2, 2, 2, 8, 13, 5, 3, 13, 6, 4, 9...
## $ quantity              <dbl> 151, 151, 39, 61, 5, 25, 8, 15, 136, 3, 16...
## $ avg__price            <dbl> 12.00000, 11.00000, 23.99000, 13.00000, 11...
## $ avg__qty              <dbl> 75.500000, 75.500000, 19.500000, 30.500000...
## $ product_refund_amount <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ cart_to_detail_rate   <dbl> 0.0519480519, 0.1724137931, 0.1989795918, ...
## $ buy_to_detail_rate    <dbl> 0.025974026, 0.008620690, 0.005102041, 0.0...
## $ year                  <dbl> 2019, 2019, 2019, 2019, 2019, 2019, 2019, ...
```

```r
products_tbl %>%
  group_by(month_of_year) %>%
  summarise(quantity = sum(quantity)) %>%
  arrange(desc(quantity)) %>%
  slice(1) %>%
  pull(month_of_year)
```

```
## [1] "2019-10-31"
```

### 4.b. Vilken produkt har sålt för mest under hela 2019?

```r
products_tbl %>%
  group_by(product) %>%
  summarise(product_revenue = sum(product_revenue)) %>%
  arrange(desc(product_revenue)) %>%
  slice(1) %>%
  pull(product)
```

```
## [1] "Google Utility BackPack"
```

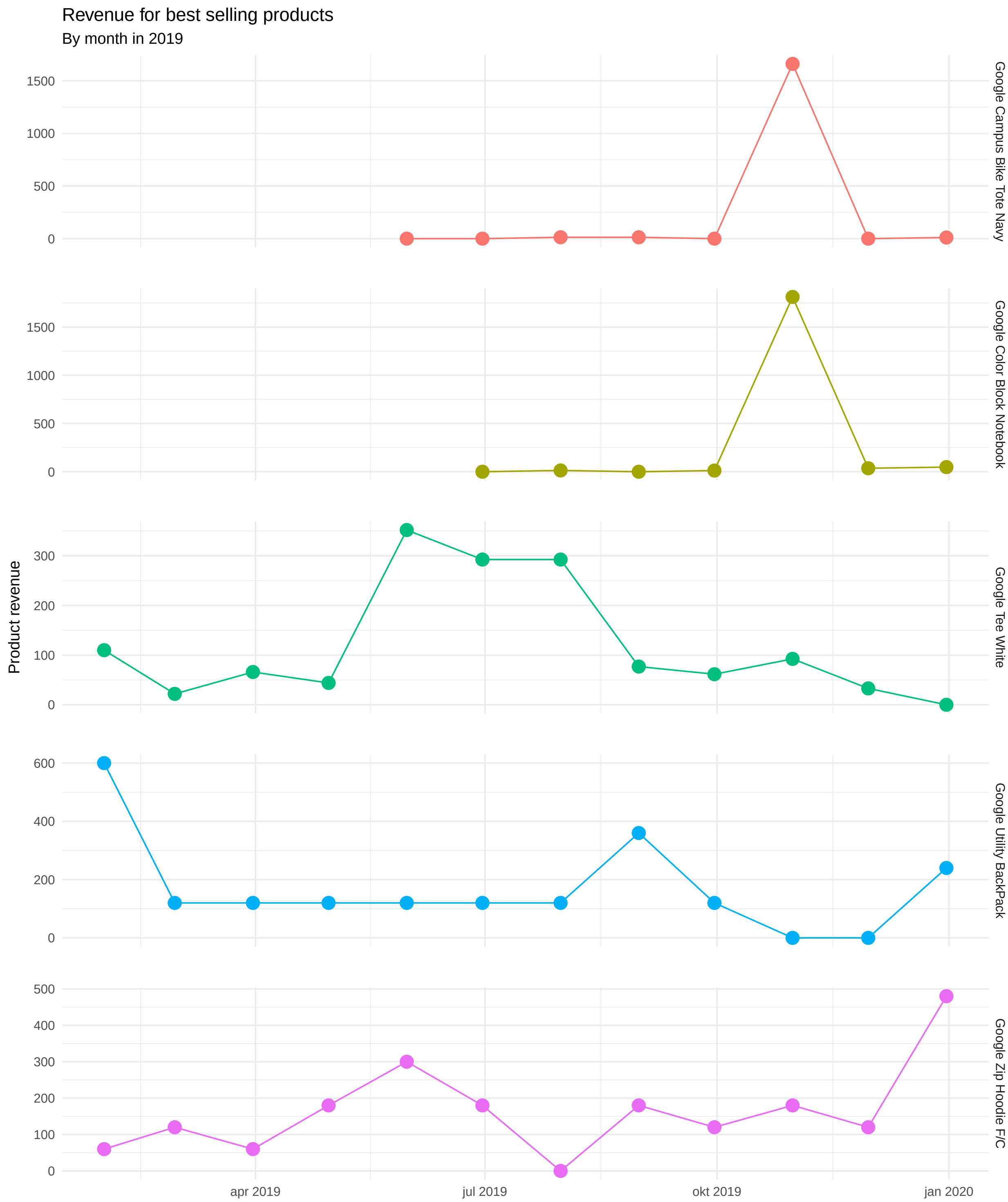
### 4.c. Visualisera försäljningsvärde för de 5 bäst säljande produkterna per månad under 2019.

```r
products_tbl %>%
  group_by(product) %>%
  summarise(product_revenue = sum(product_revenue)) %>%
  ungroup() %>%
  top_n(5, product_revenue) %>%
  pull(product) -> best_selling_products

best_selling_products
```

```
## [1] "Google Campus Bike Tote Navy" "Google Color Block Notebook"
## [3] "Google Tee White"             "Google Utility BackPack"
## [5] "Google Zip Hoodie F/C"
```

```r
products_tbl %>%
  filter(product %in% best_selling_products) %>%
  ggplot(aes(x = month_of_year, y = product_revenue, color = product)) +
  geom_point(size = 4) +
  geom_line() +
  facet_grid(rows = vars(product), scales = "free") +
  labs(
    x = "",
    y = "Product revenue",
    title = "Revenue for best selling products",
    subtitle = "By month in 2019"
```

```r
) +
theme_minimal() +
theme(panel.spacing = unit(2, "lines"),
      legend.position = "none")
```



### 5. *I mån av tid* - Vilka tre produkter har sålt flest (antal) per trafikkälla och för hur mycket har de sålt? Försök visualisera resultatet.

```
read_csv("product_medium_tbl.csv") -> product_medium_tbl

product_medium_tbl %>% glimpse()
```

```
## Observations: 2,319
## Variables: 10
## $ product                  <chr> "Google Zip Hoodie F/C", "Google Utilit...
## $ default_channel_grouping <chr> "Organic Search", "Organic Search", "Or...
## $ product_revenue          <dbl> 1439.5734, 1233.9372, 1111.7518, 1024.4...
## $ unique_purchases         <dbl> 24, 10, 60, 5, 58, 26, 50, 31, 19, 21, ...
## $ quantity                 <dbl> 24, 10, 63, 45, 63, 26, 77, 31, 19, 22,...
## $ avg__price               <dbl> 59.982223, 123.393716, 17.646853, 22.76...
## $ avg__qty                 <dbl> 1.000000, 1.000000, 1.050000, 9.000000,...
## $ product_refund_amount    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ cart_to_detail_rate      <dbl> 0.14025438, 0.03776224, 0.14891975, 0.1...
## $ buy_to_detail_rate       <dbl> 0.008250258, 0.001075847, 0.009259259, ...
```

```
product_medium_tbl %>%
  dplyr::select(-default_channel_grouping) %>%
  colnames() -> nest_vars

nest_vars
```

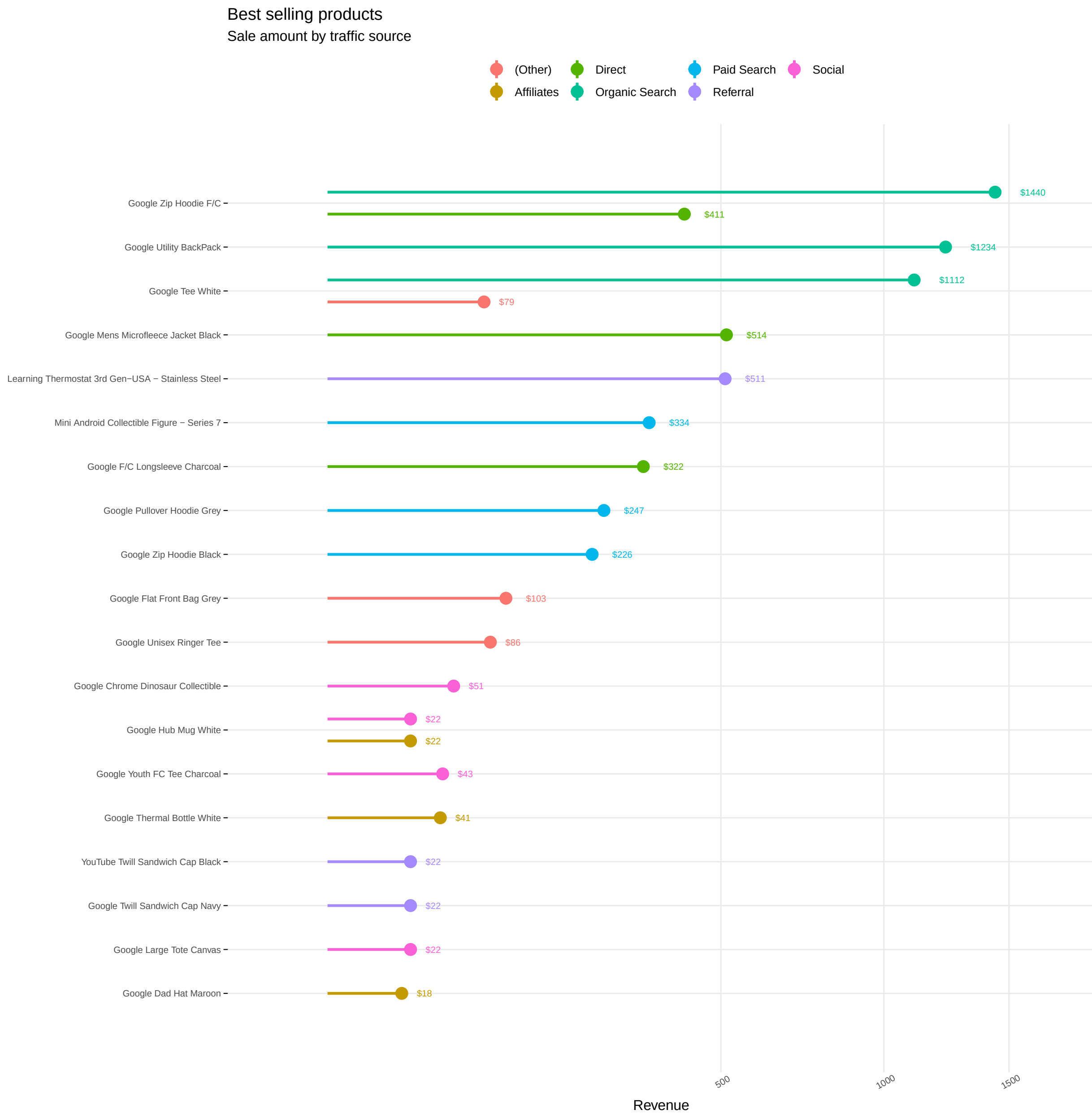
```
## [1] "product"               "product_revenue"       "unique_purchases"
## [4] "quantity"              "avg__price"            "avg__qty"
## [7] "product_refund_amount" "cart_to_detail_rate"   "buy_to_detail_rate"
```

```
select_top_by_group <- function(x) {

  x %>%
    group_by(product) %>%
    summarise(quantity = sum(quantity),
              product_revenue = sum(product_revenue)) %>%
    arrange(desc(quantity)) %>%
    top_n(3, product_revenue )

}

product_medium_tbl %>%
  nest(data = one_of(nest_vars)) %>%
    mutate(
      top_product = map(data, select_top_by_group)
    ) %>%
  unnest(top_product) %>%
  filter(quantity > 0) %>%
  dplyr::select(-data) -> sales_data
```

```r
sales_data %>% glimpse()
```

```
## Observations: 22
## Variables: 4
## $ default_channel_grouping <chr> "Organic Search", "Organic Search", "Or...
## $ product                  <chr> "Google Tee White", "Google Zip Hoodie ...
## $ quantity                 <dbl> 63, 24, 10, 10, 7, 7, 3, 2, 2, 5, 5, 2,...
## $ product_revenue          <dbl> 1111.75176, 1439.57336, 1233.93716, 322...
```

```r
    sales_data %>%
      group_by(product) %>%
      mutate(tot_rev = sum(product_revenue)) %>%
      ggplot(aes(x = reorder(product,tot_rev),
                 y = product_revenue, fill = default_channel_grouping,
                 color = default_channel_grouping)) +
      geom_point(size = 4,position = position_dodge(width = 1)) +
      geom_linerange(aes(x = reorder(product,tot_rev),
                         ymin = 0, ymax = product_revenue,
                         color = default_channel_grouping),
                     position = position_dodge(width = 1), size = 1) +
      geom_text(aes(label = product_revenue %>%
                      round(.,.2) %>%
                      paste0("$", .), color = default_channel_grouping),
                position = position_dodge(width = 1),
                hjust = -1,
                size = 2.5) +
      scale_y_sqrt(expand = c(0.15,0)) +
      scale_x_discrete(expand = c(0.1, 0)) +
      labs(
        x = "",
        y = "Revenue",
        title = "Best selling products",
        subtitle = "Sale amount by traffic source",
        fill = "",
        color = ""
      ) +
      theme_minimal() +
      coord_flip() +
      theme(axis.text.x = element_text(angle = 30, size = 7),
            legend.position = "top",
            axis.ticks.y = element_line(),
            panel.grid.minor = ggplot2::element_blank(),
            axis.text.y = element_text(size = 7),
            panel.spacing = unit(1, "lines")
      )
```




## 6. Del 2 - Estimera effekten av en marknadsföringskampanj med Causal Impact

I denna övning ska du estimera effekten av en marknadsföringskampanj som startade den 1 December 2019 och varade månaden ut. Målet med kampanjen var att öka trafiken (och i slutändan försäljningen) från betalda trafikkällor. Din uppgift är att estimera hur många extra sessioner som kampanjen genererade under december månad från början till slut.

Till hjälp har vi R-paketet *Causal Impact*. Skumma gärna igenom texten i följande artikel innan du börjar: https://google.github.io/CausalImpact/CausalImpact.html

Det data vi har till förfogande importeras först genom koden nedan.

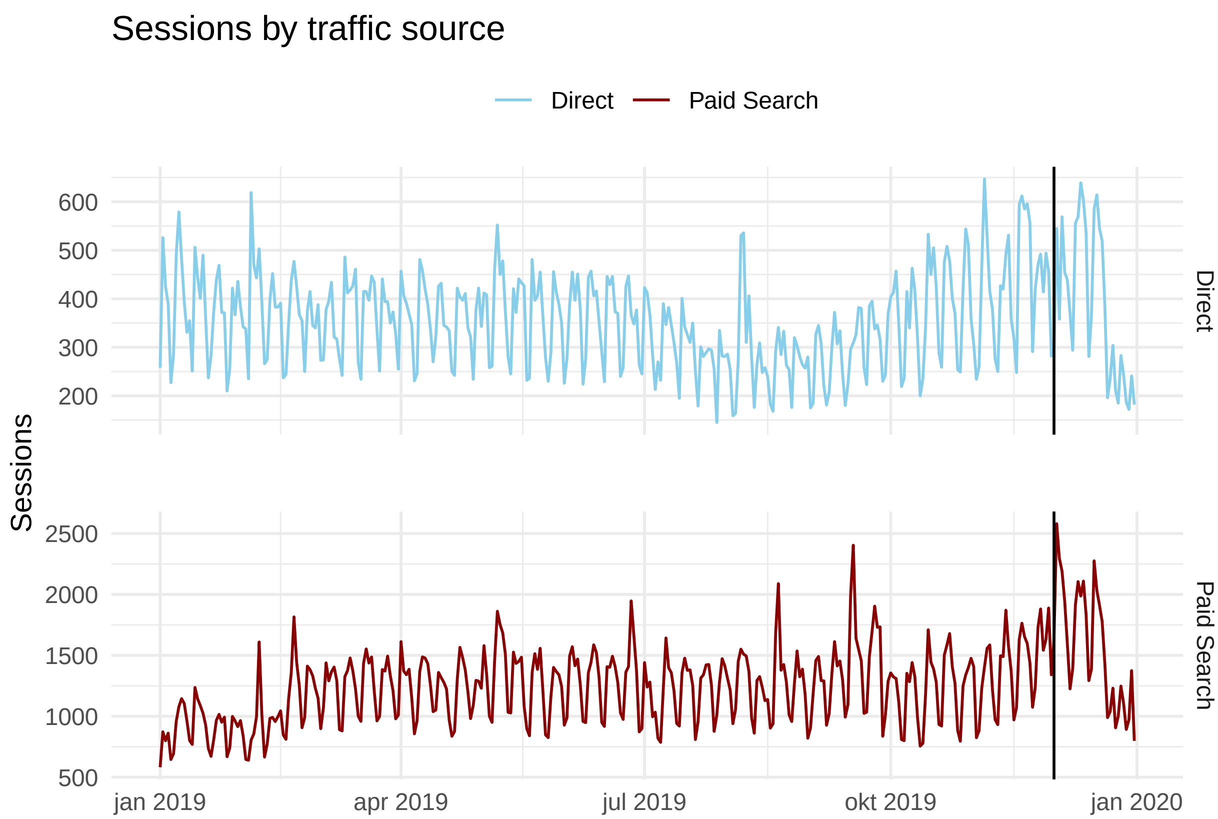
```r
read_csv("traffic_long.csv") -> traffic_long

traffic_long %>% glimpse()
```

```
## Observations: 2,896
## Variables: 11
## $ default_channel_grouping  <chr> "Paid Search", "Paid Search", "Paid Se...
## $ date                      <date> 2019-09-17, 2019-08-20, 2019-09-16, 2...
## $ users                     <dbl> 2221, 1971, 1862, 1828, 1814, 1783, 17...
## $ new_users                 <dbl> 1938, 1737, 1576, 1463, 1527, 1429, 13...
## $ sessions                  <dbl> 2404, 2089, 1997, 2581, 1889, 1947, 19...
## $ bounce_rate               <dbl> 0.5836106, 0.6108186, 0.5818728, 0.547...
## $ pages_per_session         <dbl> 3.478369, 3.037817, 3.026540, 3.468829...
## $ avg__session_duration     <dbl> 125.72546, 98.64146, 109.88232, 142.53...
## $ ecommerce_conversion_rate <dbl> 0.0012479201, 0.0004786979, 0.00100150...
## $ transactions              <dbl> 3, 1, 2, 3, 3, 3, 5, 2, 4, 2, 0, 5, 4,...
## $ revenue                   <dbl> 87.00, 29.00, 73.70, 73.80, 139.00, 18...
```

Då vi vill veta hur många sessioner från betald trafik som har tillkommit är sessioner från *Paid Search* vår **event-variabel**. Modellen kräver att vi också anger en **kontrollvariabel** som ska ha varit opåverkad av kampanjen. Eftersom direkt trafik (*direct*) bör vara opåverkad använder vi den som kontroll.

Sedan behöver vi filtrera och transformera vår data. Vi plockar först ut sessioner från de relevanta källorna med koden nedan.

```r
traffic_long %>%
  dplyr::select(default_channel_grouping, date, sessions) %>%
  filter(default_channel_grouping %in% c("Paid Search", "Direct")) -> traffic_data

traffic_data %>%
  ggplot(aes(x = date, y = sessions,
             color = default_channel_grouping)) +
  geom_line() +
  geom_vline(xintercept = as.Date("2019-12-01")) +
  labs(
    color = "",
    x = "",
    y = "Sessions",
    title = "Sessions by traffic source"
  ) +
  scale_color_manual(values = c("skyblue", "dark red")) +
  facet_grid(rows = vars(default_channel_grouping), scales = "free") +
  theme_minimal() +
  theme(legend.position = "top",
        panel.spacing = unit(2, "lines"))
```




Därefter måste vi transformera datat så att det kan hanteras av *CausalImpact()* - funktionen. Det gör vi genom att använda *spread* funktionen från *tidyr*. Vi lägger även till ett index då det behövs senare.

```
traffic_data %>% glimpse()
```

```
## Observations: 730
## Variables: 3
## $ default_channel_grouping <chr> "Paid Search", "Paid Search", "Paid Sea...
## $ date                     <date> 2019-09-17, 2019-08-20, 2019-09-16, 20...
## $ sessions                 <dbl> 2404, 2089, 1997, 2581, 1889, 1947, 190...
```

```
traffic_data %>%
  tidyr::spread(default_channel_grouping, sessions) %>%
  rename_all(.funs = function(x) x %>%
               tolower %>%
               gsub(" ", "_",.)) %>%
  arrange(date) %>%
  mutate(index = row_number())  -> traffic_wide_tbl

traffic_wide_tbl %>% glimpse()
```

```
## Observations: 365
## Variables: 4
## $ date         <date> 2019-01-01, 2019-01-02, 2019-01-03, 2019-01-04, 201...
## $ direct       <dbl> 258, 526, 425, 390, 227, 286, 490, 579, 482, 394, 33...
```

```
## $ paid_search <dbl> 582, 873, 799, 862, 645, 693, 958, 1079, 1145, 1105,...
## $ index       <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1...
```

Sedan behöver vi göra ytterligare ett par transformationer så att vårt data kan hanteras av *CausalImpact()* - funktionen.

Först definierar vi datumet som kampanjen startade för att kunna få ut motsvarande numeriska index från vårt data. Det gör vi för att kunna berätta för CausalImpact vilka index som tillhör *pre-perioden* (i.e. före kampanjen) och *post-perioden* (efter kampanjen).

```r
intervention_period = as.Date("2019-12-01")

traffic_wide_tbl %>%
  filter(date == intervention_period ) %>%
  pull(index) -> intervention_index

traffic_wide_tbl %>%
  tail(1) %>% pull(index) -> end_index

c(1,(intervention_index - 1)) -> pre_period
c(intervention_index,end_index) -> post_period
```

Nu kan vi skapa modellen efter att ha konverterat vår *tibble* med data till *matrix* format.

```r
set.seed(1)

cbind(y = traffic_wide_tbl %>% pull(paid_search),
      x1 = traffic_wide_tbl %>% pull(direct)) -> int_data

class(int_data)
```

```
## [1] "matrix"
```

```r
impact <- CausalImpact(int_data, pre_period, post_period)

summary(impact)
```

```
## Posterior inference {CausalImpact}
##
##                          Average        Cumulative
## Actual                   1563           48465
## Prediction (s.d.)        1356 (65)      42037 (2016)
## 95% CI                   [1226, 1490]   [38009, 46175]
##
## Absolute effect (s.d.)   207 (65)       6428 (2016)
## 95% CI                   [74, 337]      [2290, 10456]
##
## Relative effect (s.d.)   15% (4.8%)     15% (4.8%)
## 95% CI                   [5.4%, 25%]    [5.4%, 25%]
##
## Posterior tail-area probability p:   0.00111
## Posterior prob. of a causal effect:  99.88864%
##
## For more details, type: summary(impact, "report")
```

Som rapporten ovan berättar är sannolikheten för en kausal effekt 99.9%.

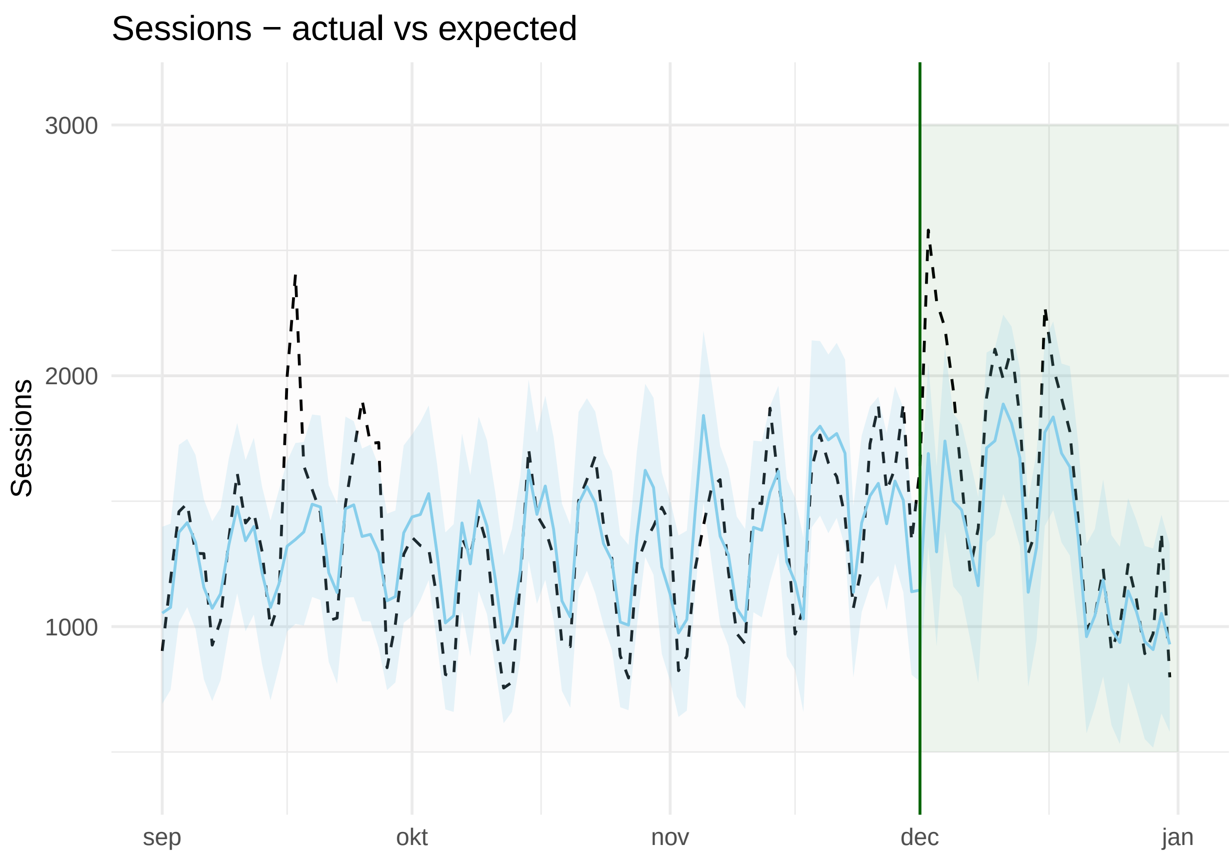
Vi kommer åt resultatet genom *impact*-objektet.

```
impact$series %>% as_tibble() %>%
    mutate(date = seq(as.Date("2019-01-01"), as.Date("2019-12-31"), "day")) %>%
    dplyr::select(date, everything()) -> ci_res

ci_res %>% glimpse()
```

```
## Observations: 365
## Variables: 15
## $ date               <date> 2019-01-01, 2019-01-02, 2019-01-03, 2019-01-...
## $ response           <dbl> 582, 873, 799, 862, 645, 693, 958, 1079, 1145...
## $ cum.response       <dbl> 582, 1455, 2254, 3116, 3761, 4454, 5412, 6491...
## $ point.pred         <dbl> 643.5173, 1205.9472, 996.3451, 924.9980, 586....
## $ point.pred.lower   <dbl> 248.2124, 827.2448, 644.7418, 544.8428, 225.7...
## $ point.pred.upper   <dbl> 1003.6882, 1565.2361, 1367.9166, 1304.5152, 9...
## $ cum.pred           <dbl> 582, 1455, 2254, 3116, 3761, 4454, 5412, 6491...
## $ cum.pred.lower     <dbl> 582, 1455, 2254, 3116, 3761, 4454, 5412, 6491...
## $ cum.pred.upper     <dbl> 582, 1455, 2254, 3116, 3761, 4454, 5412, 6491...
## $ point.effect       <dbl> -61.51734, -332.94720, -197.34506, -62.99798,...
## $ point.effect.lower <dbl> -421.6882, -692.2361, -568.9166, -442.5152, -...
## $ point.effect.upper <dbl> 333.78762, 45.75521, 154.25815, 317.15716, 41...
## $ cum.effect         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ cum.effect.lower   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ cum.effect.upper   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
```

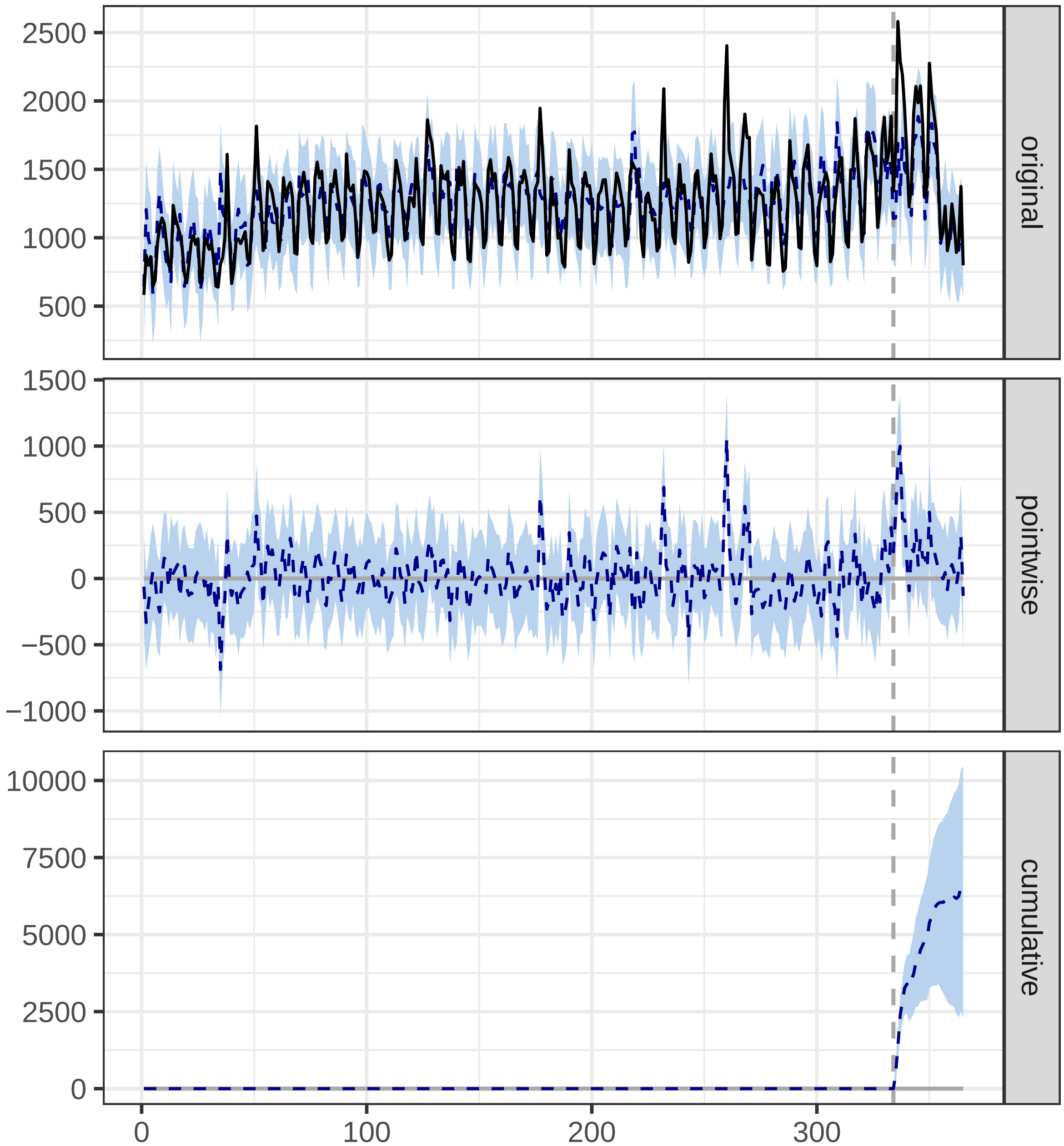
Vi kan visualisera resultatet genom:

```
ci_res %>%
  filter(date >= as.Date("2019-09-01")) %>%
  ggplot(aes(date, y = response)) +
  geom_line(aes(colour = "blue"),color = "black", linetype = 2, show.legend = T) +
  annotate("rect", xmin = as.Date("2019-12-01"),
                xmax = as.Date("2020-01-01"),
                 ymin =500, ymax = 3000 ,
           alpha = .07, fill = "dark green") +
    annotate("rect", xmin = as.Date("2019-09-01"),
                xmax = as.Date("2019-12-01"),
                 ymin =500, ymax = 3000 ,
           alpha = .01, fill = "dark red") +
  geom_line(aes(date, y = point.pred), color = "skyblue", show.legend = T) +
  geom_ribbon(aes(xmin = date, xmax = date,
                 ymin =point.pred.lower,
                ymax = point.pred.upper ), alpha = 0.2, fill = "skyblue") +
  scale_y_continuous(expand = c(0.1,0)) +
  geom_vline(xintercept = as.Date("2019-12-01"), color = "dark green") +
  labs(
    x = NULL,
    y = "Sessions",
    title = "Sessions - actual vs expected"
  ) +
  theme_minimal()
```




Det finns även en inbyggd plotfunktion för en snabbare visualisering.

```
plot(impact)
```




Med modellen skapad är vi nu redo att svara på övningsfrågorna.

### 2.a. Hur många sessioner observerades totalt mellan 2019-12-01 - 2019-12-31?

```
traffic_wide_tbl %>%
  filter(date >= as.Date("2019-12-01") &
           date <= as.Date("2019-12-31")) %>%
  summarise(paid_search = sum(paid_search)) %>%
  pull(paid_search)
```

```
## [1] 48465
```

### 2.b. Gav kampanjen en signifikant effekt avseende ökning i antal sessioner?

```r
impact$summary %>%
  t() %>% data.frame() %>%
  rownames_to_column("variable") %>%
  as_tibble() -> impact_results

impact_results
```

```
## # A tibble: 15 x 3
##    variable           Average  Cumulative
##    <chr>                <dbl>       <dbl>
##  1 Actual           1563.      48465
##  2 Pred             1356.      42037.
##  3 Pred.lower       1226.      38009.
##  4 Pred.upper       1490.      46175.
##  5 Pred.sd            65.0      2016.
##  6 AbsEffect         207.       6428.
##  7 AbsEffect.lower    73.9      2290.
##  8 AbsEffect.upper   337.      10456.
##  9 AbsEffect.sd       65.0      2016.
## 10 RelEffect           0.153       0.153
## 11 RelEffect.lower     0.0545      0.0545
## 12 RelEffect.upper     0.249       0.249
## 13 RelEffect.sd        0.0480      0.0480
## 14 alpha               0.05        0.05
## 15 p                   0.00111     0.00111
```

```r
impact_results %>%
  filter(variable == "p" | variable == "alpha") %>%
  pull(Cumulative) %>%
  round(.,3)
```

```
## [1] 0.050 0.001
```

### 2.c. Hur många sessioner kan krediteras till kampanjen?

```r
impact_results %>%
  filter(variable == "AbsEffect") %>%
  pull(Cumulative) %>% round() -> session_effect

session_effect
```

```
## [1] 6428
```

### 2.d. Vad var den totala procentuella ökningen i sessioner?

```
impact_results %>%
  filter(variable == "RelEffect") %>%
  mutate_at(vars(Cumulative), function(x) x*10^2) %>%
  pull(Cumulative) %>%
  round(.,1) %>%
  paste0(., "%")
```

```
## [1] "15.3%"
```

### Extraupgift. Kampanjen kostade $1000 att genomföra. Var det en bra investering?

*Ledtråd - du räknade ut konverteringsgraden för betald trafik i uppgift 2 c). Du kan även tänkas behöva genomsnittligt transaktionsvärde för att ta reda på snittvärdet av en session från betald trafik.*

```
paid_search_conversion_rate = 0.0032

    traffic_source_tbl %>%
      group_by(default_channel_grouping) %>%
      summarise(transactions = sum(transactions),
                revenue = sum(revenue)) %>%
      mutate(revenue_per_transaction = revenue / transactions) %>%
      filter(default_channel_grouping == "Paid Search") %>%
      pull(revenue_per_transaction) %>%
      round() -> paid_search_avg_revenue

session_effect*
  paid_search_conversion_rate*
    paid_search_avg_revenue -> estimated_value_of_campaign

paste0("$",estimated_value_of_campaign %>% round)
```

```
## [1] "$1131"
```